

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.

2/5/1 (Item 1 from file: 351)
DIALOG(R)File 351:Derwent WPI
(c) 2004 Thomson Derwent. All rts. reserv.

012648101 **Image available**
WPI Acc No: 1999-454206/ 199938
XRPX Acc No: N99-340240

CAD system for electronic system design - has memory to store design information by data structure which can preserve several detailed information based on level layer
Patent Assignee: FUJITSU LTD (FUIT)
Inventor: KOBAYASHI Y; KOBUCHI E; KUBOTA K; NAKAMURA T; OHTSUKA M; SAITO H T
Number of Countries: 002 Number of Patents: 002
Patent Family:
Patent No Kind Date Applcat No Kind Date Week
JP 11191116 A 19990713 JP 9872920 A 19980320 199938 B
US 6304790 B1 20011016 US 98177301 A 19981022 200164

Priority Applications (No Type Date): JP 97290773 A 19971023

Patent Details:

Patent No	Kind	Lan Pg	Main IPC	Filing Notes
JP 11191116	A	25	G06F-017/50	
US 6304790	B1		G06F-019/00	

Abstract (Basic): JP 11191116 A

NOVELTY - A parameter setting unit (22) sets the design information to the symbols and the connecting lines edited from block diagram contains symbol. Each setting information has memory to store the design information using a data structure which can preserve several detailed level information, based on a level layer. DETAILED DESCRIPTION - An editor (2) edits the block diagram which contains symbols representing independent functions and the lines connecting the symbols. An estimated parameter information provided on the symbols or the line is processed and stored in a template. Based on the template information, parameter is demanded from the user. An INDEPENDENT CLAIM is also included for a program memory medium.

USE - In system designing of electronic apparatus.

ADVANTAGE - Optimum component design is obtained at early stages of design by performing high precision estimation of component parameters. Design techniques and estimation parameters are stored in a library and hence a re-entry of process and a design mistake is prevented, thereby improves estimation accuracy. DESCRIPTION OF DRAWING(S) - The figure is block diagram showing the components of CAD system. (2) Editor; (22) Parameter setting unit.

Dwg.1/36

Title Terms: CAD; SYSTEM; ELECTRONIC; SYSTEM; DESIGN; MEMORY; STORAGE; DESIGN; INFORMATION; DATA; STRUCTURE; CAN; PRESERVE; DETAIL; LEVEL; INFORMATION; BASED; LEVEL; LAYER

Derwent Class: T01

International Patent Class (Main): G06F-017/50; G06F-019/00

File Segment: EPI

2/5/2 (Item 1 from file: 347)
DIALOG(R)File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

06249539 **Image available**
SYSTEM DESIGN AND EVALUATION CAD SYSTEM AND PROGRAM STORAGE MEDIUM THEREFOR

PUB. NO.: 11-191116 A]
PUBLISHED: July 13, 1999 (19990713)
INVENTOR(s): NAKAMURA TAKEO
TAKABUCHI HIDEKO
OTSUKA MASATO
SAITO TAKASHI

KOBAYASHI YUKIYASU
KUBOTA KENICHI
APPLICANT(s): FUJITSU LTD
APPL. NO.: 10-072920 [JP 9872920]
FILED: March 20, 1998 (19980320)
PRIORITY: 09290773 [JP 979290773], JP (Japan), October 23, 1997
(19971023)
INTL CLASS: G06F-017/50

ABSTRACT

PROBLEM TO BE SOLVED: To provide a system, with which high-accuracy and high-reality estimation is enabled in an upstream process (on the early stage of design) and further design information can be succeeded to a downstream process, concerning a CAD system for supporting the system configuration of an electronic device or the design check of an architecture.

SOLUTION: A block drawing inputting/editing part 2 edits a block drawing containing symbols expressing independent functions and lines connecting these symbols. These symbols and lines similarly represent the design information and design information or parameter information can be similarly applied. At the same time, this system has structure for providing plural pieces of detailed level information in every design information as the image of layers. Besides, a parameter template holding part 5 prepares estimate parameter information to be applied to the symbols and lines as a template so that the information required for a user can be inputted.

COPYRIGHT: (C)1999, JPO

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-191116

(43)公開日 平成11年(1999)7月13日

(51)Int.Cl.⁶

G 0 6 F 17/50

識別記号

F I

G 0 6 F 15/60

6 5 4 K

審査請求 未請求 請求項の数13 OL (全25頁)

(21)出願番号 特願平10-72920

(22)出願日 平成10年(1998)3月20日

(31)優先権主張番号 特願平9-290773

(32)優先日 平9(1997)10月23日

(33)優先権主張国 日本 (JP)

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番

1号

(72)発明者 中村 武雄

神奈川県川崎市中原区上小田中4丁目1番

1号 富士通株式会社内

(72)発明者 ▲高▼渕 英子

神奈川県川崎市中原区上小田中4丁目1番

1号 富士通株式会社内

(74)代理人 弁理士 遠山 勉 (外1名)

最終頁に続く

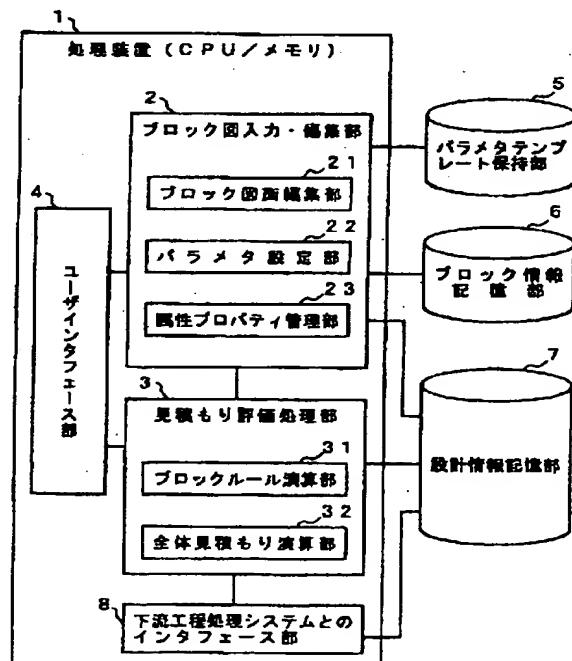
(54)【発明の名称】 システム設計／評価CADシステムおよびそのプログラム記憶媒体

(57)【要約】

【課題】電子装置のシステム構成やアーキテクチャの設計検討を支援するCADシステムに関し、上流工程（設計の早期段階）での高精度で実現性の高い見積もりを可能とし、取得した設計情報をノウハウとして蓄積し、さらに下流工程へ設計情報を継承できるシステムを実現する。

【解決手段】ブロック図入力・編集部により、独立した機能を表現するシンボルとそれらのシンボルをつなぐラインとを含むブロック図を編集する。これらのシンボルとラインは同じように設計情報を代表したものとし、どちらも同様に設計情報やパラメタ情報を付与可能にする。かつ、それぞれの設計情報にレイヤという考え方で複数の詳細レベル情報を持たせる構造とする。また、シンボル、ラインに付与する見積もりパラメタ情報を、テンプレートとしてパラメタテンプレート保持部5に用意し、これによってユーザに必要な情報を入力させる。

本発明のブロック構成例を示す図



【特許請求の範囲】

【請求項1】 コンピュータによりシステム設計を支援するCADシステムにおいて、

独立した機能を表現するシンボルとそれらのシンボルをつなぐラインとを含むブロック図を編集する手段と、前記シンボルと前記ラインとに、それぞれ同じ形態で設計情報を設定する手段と、

それぞれの設定情報にレベルレイヤに応じた複数の詳細レベル情報を持たせることができるデータ構造で設計情報を記憶する手段とを備えたことを特徴とするシステム設計／評価CADシステム。

【請求項2】 請求項1記載のシステム設計／評価CADシステムにおいて、前記ブロック図を編集する手段により入力するシンボルまたはラインに付与する見積もりパラメタ情報を入力するためのテンプレートを保持する手段と、

前記テンプレートに従ってユーザに入力を促し、シンボルまたはラインに付与する見積もりパラメタ情報をユーザから入力する手段とを備えたことを特徴とするシステム設計／評価CADシステム。

【請求項3】 請求項1記載のシステム設計／評価CADシステムにおいて、

設計の途中段階でその設計の評価を行った場合に、評価を行った時点の設計情報と評価結果情報をリンクして自動保管する手段を備えたことを特徴とするシステム設計／評価CADシステム。

【請求項4】 請求項1記載のシステム設計／評価CADシステムにおいて、

再利用可能な設計ブロックを利用する際に、そのブロックの持つパラメタテンプレートを参照し、パラメタテンプレートに基づくパラメタ設定ウインドウを表示する手段と、

表示したパラメタ設定ウインドウにより、当該ブロック利用時の固有なパラメタを入力する手段とを備えたことを特徴するシステム設計／評価CADシステム。

【請求項5】 請求項1記載のシステム設計／評価CADシステムにおいて、

設計の詳細化が進むにつれて一つのブロックが、より詳細なレベルの複数のブロックから構成されるようになった場合に、重複するパラメタ情報はより詳細な下位の設定レベルに付与されたパラメタを利用して評価を行う手段を備えたことを特徴とするシステム設計／評価CADシステム。

【請求項6】 コンピュータによりシステム設計を支援するCADシステムに用いられるプログラムを記憶した媒体であって、

独立した機能を表現するシンボルとそれらのシンボルをつなぐラインとを含むブロック図を編集する処理と、

前記シンボルと前記ラインとに、それぞれ同じ形態で設計情報を設定する処理と、

それぞれの設定情報にレベルレイヤに応じた複数の詳細レベル情報を持たせることができるデータ構造で設計情報を管理する処理とをコンピュータに実行させるプログラムを記憶したことを特徴とするシステム設計／評価CADシステムのプログラム記憶媒体。

【請求項7】 請求項1記載のシステム設計／評価CADシステムにおいて、前記レベル情報設定手段は、情報はレベルレイヤに応じた複数レベルの情報を参照しながらその設計情報の編集を行う設計情報編集手段を備えたことを特徴とするシステム設計／評価CADシステム。

【請求項8】 請求項1記載のシステム設計／評価CADシステムにおいて、前記設計情報に、ドキュメントデータをリンクするドキュメントデータリンク手段を備え、設計情報の編集中にドキュメントデータの編集を行うドキュメントデータ編集手段を備えたことを特徴とするシステム設計／評価CADシステム。

【請求項9】 請求項1記載のシステム設計／評価CADシステムにおいて、前記設計情報設定手段は、前記シンボルの大きさを任意の大きさに設定する大きさの編集手段を有することを特徴とするシステム設計／評価CADシステム。

【請求項10】 請求項1記載のシステム設計／評価CADシステムにおいて、前記ブロック図編集手段は、作成したシンボルにラインを接続したとき、信号の入出力ポートを自動的に作成する入出力ポート作成手段を有することを特徴とするシステム設計／評価CADシステム。

【請求項11】 請求項1記載のシステム設計／評価CADシステムにおいて、一つのシンボルに他の設計情報をリンクさせて複数の階層情報を割り付ける手段を備え、一つのシンボルに複数の階層を定義することを特徴とするシステム設計／評価CADシステム。

【請求項12】 請求項1記載のシステム設計／評価CADシステムにおいて、シンボル作成用記述言語及びブロック図のデータをシンボル作成パネル上にドラッグ＆ドロップする事によって、シンボル形状データを自動的に作成するシンボル自動作成手段を備えたことを特徴とするシステム設計／評価CADシステム。

【請求項13】 請求項1記載のシステム設計／評価CADシステムにおいて、複数のブロックを指定して、それらを一つの階層に合成あるいは複数の階層に分離する階層編集手段を備えたことを特徴とするシステム設計／評価CADシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は電子装置のシステム設計を支援するCADシステムに係わり、特に装置の仕様検討やアーキテクチャの検討を行う設計フェーズへの利用を図り、装置設計全体の設計最適化、設計品質の向上、設計の効率化を実現するシステム設計／評価CAD

システムおよびそのプログラム記憶媒体に関するものである。

【0002】

【従来の技術】従来、電子装置の設計の初期段階、つまり仕様やアーキテクチャの検討段階において、装置を構成するLSIやプリント回路基板(PCB)回路の物理的な実現規模や消費電力などを見積もるには、過去に設計した装置、PCB、LSIの同様な機能などを参考にし、机上で個々の構成要素の物理的な実現規模等を見積もり、それを装置に組み上げたときの規模などを算出していた。したがって、設計者の経験によるところが大きかった。

【0003】ここで見積もる物理的なパラメタには次のようなものがある。LSIやFPGAなどではゲート数(面積)、チップの入出力となるピン数、チップの消費電力、遅延(サイクル数)、PCBに搭載する際の実装面積、PCBへの搭載部品数、PCBの入出力となるコネクタピン数などである。また個々の部品の価格情報なども見積もり評価のためのパラメタとなる。

【0004】例えば図21に示すようなブロック図を用いて装置の機能構成を検討し、一つの機能を表す個々のブロックである箱(シンボル)について、前述の物理パラメタを見積もり値として積算、集計する。ブロック図における一つの箱(シンボル)は中身をさらに詳細化し、同様のブロック図で表現されるような階層化が行われる。この場合、下位の階層においても個々の箱について見積もりを行い、集計結果を上位の箱に積み上げるという方法になる。

【0005】もちろん、個々の機能が過去に設計済みである箱(シンボル)を流用するのであれば、その情報から正確な見積もり結果を得ることができる。装置設計の初期検討段階では、装置の機能・性能と共に、物理的な実現性(部品点数、大きさ、熱など)やコストも重要な検討要素であり、装置のアーキテクチャ、LSIやPCBの分割方法などを変えた数多くのパターンで最適な設計を確認する必要がある。

【0006】従来は図36のようなブロック図を書き、各ブロックのパラメタ値を見積もり、それらを入手、あるいは表計算ソフトウェアなどを使って集計していた。

【0007】

【発明が解決しようとする課題】前述のように従来の手法では、装置設計者の経験や勘により個々のブロックの見積もりを行い、それを積み上げていた。このため、以下に挙げるような問題点があった。

【0008】・設計者個人の経験による見積もりであるため、見積もり精度に個人差が大きく現れる。

・設計・見積もりのノウハウが個人に留まることになり、共通情報として蓄積されない。

【0009】・装置の構成をカット・アンド・トライする場合に、煩雑な同様の計算を繰り返し行う必要があ

り、時間を要する。また計算ミスも増える。

・求めた装置構成やアーキテクチャが最適な解であるとの客観的な評価が難しい。

【0010】・仕様検討の段階で作成した装置構成ブロック図などの設計情報が以降の設計に活かせず、下流工程でCADシステムへの再入力が必要になる。本発明は上記問題点の解決を図り、装置設計の上流工程であるシステム設計の工程をコンピュータで支援する手段を提供することを目的とする。

10 【0011】

【課題を解決するための手段】図1は、本発明のブロック構成例を示す図である。図1において、1はCPUおよびメモリなどからなる処理装置、2はブロック図入力・編集部、3は見積もり評価処理部、4はユーザインターフェース部、5はパラメタテンプレート保持部、6はブロック情報記憶部、7は設計情報記憶部、8は下流工程処理システムとのインターフェース部を表す。

【0012】ブロック図入力・編集部2は、ブロック図を入力・編集し、編集結果のデータをブロック情報記憶部6へライブラリとして保存し、また設計情報記憶部7へブロック図とともにに入力された設計情報を格納する手段である。ブロック図入力・編集部2は、ブロック図面編集部21、パラメタ設定部22、属性プロパティ管理部23を持つ。

【0013】ブロック図面編集部21は、設計対象のブロック図を構成するシンボル、ラインを入力し編集する手段である。ここでは、ブロック図のシンボルとラインとは同じように設計情報を代表(シンボライズ)したものとして扱う。モデル(シンボル、ライン)の入力には、パラメタテンプレート保持部5に登録されたパラメタテンプレートを用いることができ、また、ブロック情報記憶部6にライブラリ化されている既存のブロック情報を利用することができる。

【0014】パラメタ設定部22は、パラメタテンプレート保持部5に記録された見積もり評価用パラメタのテンプレートにパラメタの追加または変更を行い、またパラメタの値を任意に設定する。

【0015】属性プロパティ管理部23は、ブロック図面編集部21により入力されたモデル(シンボル、ライン)に設計情報やパラメタ情報を付与し管理する手段である。具体的には、モデルに物理的パラメタ、下位のブロックや他の環境で設計した設計情報などへのリンク情報、見積もり評価用のパラメタ等の設計情報やパラメタ情報を属性プロパティとして管理する。

【0016】また、再利用が可能なシンボルやラインの属性プロパティを設計情報記憶部7に記憶してライブラリ化する。さらに、各ブロックの設計情報に複数の階層化した詳細情報を保持できるような構造を持たせて管理する。

50 【0017】見積もり評価処理部3は、ブロックごとの

見積もり評価およびブロック図全体の見積もり評価を行い、その評価結果を設計情報記憶部7へ格納する手段であり、ブロックルール演算部31、全体見積もり演算部32を持つ。

【0018】ブロックルール演算部31は、そのブロック固有のパラメタと見積もり評価ルールを用いてブロックの見積もり評価を行う手段である。全体見積もり演算部32は、各ブロックに指定されたパラメタおよびブロックルール演算部31で計算された値を用いて設計対象全体の見積もり評価を行う手段である。

【0019】ユーザインターフェース部4は、ブロック図入力・編集部2または見積もり評価処理部3とユーザとのインターフェースをとる手段である。GUI(Graphical User Interface)を用いる。

【0020】パラメタテンプレート保持部5は、ブロック図の入力・編集において、シンボル、ラインのパラメタ入力のためのテンプレートを記憶する手段である。ブロック情報記憶部6は、ブロック図入力・編集部2で生成された再利用が可能なブロック情報をライブラリ化して登録し保存する手段である。ブロック情報とは、そのブロックのシンボル、固有のプロパティテンプレート、見積もり評価ルール等である。

【0021】設計情報記憶部7は、設計情報をデータベースとして記憶する手段である。下流工程処理システムとのインターフェース部8は、本システムと下流工程の処理システムとのインターフェースをとる手段である。

【0022】本発明は、以下のような特徴を持つ。

(1) 本発明は、シンボルとラインと同じようなモデルとして扱い、また複数の抽象レベルが扱えるブロック図入力・編集システムである。

【0023】システムの設計・検討を行う際に作成するブロック図では、独立した機能を表現する箱(シンボル)と、箱と箱とをつなぐ線(ライン)自体も機能を持つ。いずれも設計情報やパラメタ情報を付与できる構造で、かつそれぞれの設計情報にレイヤという考え方で複数の詳細レベル情報を持たせることができる構造としている。

【0024】本システムは、シンボルとラインとを、同じように設計情報を代表(シンボライズ)するものとして編集する。

(2) 本発明は、プロパティとパラメタテンプレートによる汎用性を持たせたブロック図編集・評価システムである。

【0025】本システムは、パラメタテンプレート保持部5に、ブロック図入力・編集部2で入力したシンボルやラインに付与する見積もりパラメタ情報をテンプレートとして用意することで、ユーザへの入力促進を行う。さらに、設計対象や見積もり評価内容に応じた任意のパラメタを自由に追加・変更可能なパラメタ設定部22を

持つ。

【0026】(3) 本発明は、設計・評価過程の設計および評価結果データの蓄積ができるブロック図編集・評価システムである。本システムは、設計の途中段階での設計の評価を行った場合に、評価を行った時点の設計情報と評価結果情報をリンクして自動保管する見積もり評価処理部3を備える。これにより様々な設計ケースをカット・アンド・トライで評価し、多数の評価結果を比較検討し、最適解の設計を採用することを可能にする。

【0027】(4) 本発明は、設計ブロックへの評価ルールを蓄積できるブロック図編集・評価システムである。本システムは、再利用可能な設計ブロックに対して、その設計ブロックを利用する際の評価を行うための入力用のパラメタテンプレート、さらに入力されたパラメタに基づいて見積もり評価(計算)を行うための見積もり評価ルールを登録できる。

【0028】(5) 本発明は、ブロックごとのパラメタテンプレートによるパラメタ設定が可能なブロック図入力・編集システムである。本システムは、再利用可能なブロックを利用する際に、ブロック情報記憶部6に記憶したそのブロックの持つパラメタテンプレートを参照し、パラメタの設定ウィンドウを表示することにより、当該ブロック利用時の固有なパラメタの入力促進を図る。

【0029】(6) 本発明は、入力のレベルに応じた精度での評価パラメタを採用するブロック図評価システムである。本システムは、設計の詳細化が進むにつれて一つのブロックが、より詳細なレベルの複数のブロックから構成されるようになった場合には、重複するパラメタ情報はより詳細な設計レベル(下位レベル)に付与されたパラメタを利用して評価を行う全体見積もり演算部32を備える。全体見積もり演算部32は、下位レベルブロックのパラメタが不完全な場合には上位ブロックのパラメタを採用する。つまり、パラメタ毎により詳細なレベルで、かつそのレベルで完全な(そのレベルで全てのブロックに付与されている)パラメタを採用する機能を持つ。

(7) 本発明では(1)において、前記レベル情報設定手段は、情報はレベルレイヤに応じた複数レベルの情報を参照しながらその設計情報の編集を行う設計情報編集手段を備えることが好ましい。(1)において、前記設計情報に、ドキュメントデータをリンクするドキュメントデータリンク手段を備え、設計情報の編集中にドキュメントデータの編集を行うドキュメントデータ編集手段を備えることができる。(1)において、前記設計情報設定手段は、前記シンボルの大きさを任意の大きさに設定する大きさの編集手段を有することを特徴とするシステム設計/評価CADシステム。(1)において、前記ブロック図編集手段は、作成したシンボルにラインを接

続したとき、信号の入出力ポートを自動的に作成する入出力ポート作成手段を有するようにすることも可能である。。(1)において、一つのシンボルに他の設計情報をリンクさせて複数の階層情報を割り付ける手段を備え、一つのシンボルに複数の階層を定義するようにすることも可能である。(1)において、シンボル作成用記述言語及びブロック図のデータをシンボル作成パネル上にドラッグ&ドロップする事によって、シンボル形状データを自動的に作成するシンボル自動作成手段を備えたようにすることも可能である。。(1)において、複数のブロックを指定して、それらを一つの階層に合成あるいは複数の階層に分離する階層編集手段を備えたようにすることも可能である。

【0030】また、各階層において、ブロックの編集を行い、その階層での編集内容を、他の上位あるいは下位の階層に反映させることも可能である。なお、以上の各構成は可能な限り互いに組み合わせて使用することができる。

【0031】本発明は、以下のように作用する。本発明では、システム検討のためにブロック図をベースにして構成図を入力し、各シンボルやライン（インターフェース）にパラメタを付与し、見積もり評価を行う。さらに、アーキテクチャ検討で使用したブロック図は設計の詳細化に使用することができ、ブロック図が徐々に回路図レベルまで詳細化できるようになっている。またはブロック図とHDL（ハードウェア設計言語）設計データとのリンク情報を持ち、上流工程から下流工程まで扱えるエントリツールの構造となる。

【0032】

【発明の実施の形態】以下に、本発明の実施の形態の一例を説明する。

【1】ブロック図による情報エントリ

コンピュータによりシステム設計の支援を行うには、まず設計情報をコンピュータに入力する必要がある。本発明では、ブロック図という設計ドキュメントを扱い、ブロック図入力・編集部2によりブロック図を入力・編集する。

【0033】ブロック図入力・編集部2の機能は、概ね従来からなるCAD技術であるが、システム設計の段階で使用するため、柔軟性のある操作性や各種の物理的パラメタなどの情報を付与できるようにしている。

【0034】本発明によるブロック図入力・編集部2の機能の一つの特徴は、個々のシンボル間を接続するラインをシンボルと同レベルで扱い、共に階層的な表現ができるようにしている点である。従来のブロック図や回路図の入力では、シンボル（ブロック）が機能を表現し、それらを接続するラインは単なるワイヤであった。本発明のシステムで扱うブロック図では、ラインを単純なワイヤではなく、機能を持たせることができるモデルとして扱うようにする。

【0035】さらに、ラインへの物理的なパラメタも付与できるようにしている。図2は、エントリ設計モデルの考え方を説明する図である。図2に示すように、シンボルやライン（シグナル）の設計情報を属性プロパティとして管理する。シンボルの設計情報の実体は機能ブロックであり、シグナルの設計情報の実体はインターフェースである。設計情報は、それぞれのモデルの属性プロパティとして管理する。

【0036】具体的に、ラインの機能とは、プロトコルそのものであったり、プロトコルを実現する機能などである。この物理的パラメタとは、そのプロトコルを実現した場合の実際のワイヤ本数や機能を具体化するための回路規模、また信号のトグル周波数であったりする。

【0037】機能ブロックは、階層設計の下位ブロックや他の環境で設計した設計データなどへのリンク、見積もり評価用のパラメタを持ち、インターフェースも同様に設計データへのリンク情報、評価パラメタなどを持つ。

【0038】機能ブロック、インターフェースは、それぞれ抽象レベルの異なる設計情報を抽象レベルレイヤといいう概念で複数持つことができる構造とする。図3は、設定情報のレベルレイヤの例を示す図である。この例では、機能ブロックおよびインターフェースの設計情報は、レイヤ1～レイヤ3の複数の構造を持っている。レイヤ1として日本語ドキュメント（d o c）、仕様記述等の情報、レイヤ2として例えばハードウェア記述言語（HDL）等の情報、レイヤ3として実際の回路図等の情報を持つ。レイヤ1には、もっとも抽象度が高いレベルの情報が記述され、レイヤ2、レイヤ3の順に、より具体的で詳細な情報が記述されていく。

【0039】設計情報は、ブロック図で表現した機能ブロック、インターフェースのシンボルに対して抽象的な記述から具体的な詳細記述まで複数レベルの設計情報にリンク可能な構造とする。このような構造により、抽象度の高いレベルから設計を始め、段階的に詳細化する設計手法がとれるようになる。

【0040】以上説明した設計情報のレベルレイヤを管理し、設計情報を保持するために、図4に示すようなデータ構造の設計情報を持つ。図4は、設計情報のデータ構造の例を示す図である。

【0041】図4中、シンボル、ライン（シグナル）の各属性プロパティが、それぞれの設計情報へのリンクや、見積もり評価用のパラメタなど複数のパラメタを保持し、設計データ全体へのリンク構造を形成している。

【0042】例えば、シグナルの属性プロパティは、そのシグナル個別情報と次のレベルレイヤのインターフェースへのリンク情報を持つ。さらに、リンク先のインターフェースには、レイヤの情報とそのインターフェースプロパティを持ち、インターフェースプロパティは、インターフェースの設計情報へのリンク情報を持つ。

【0043】同様に、シンボルの属性プロパティは、そ

のシンボル個別情報と次のレベルレイヤのモジュールへのリンク情報を持つ。ここで、モジュールとは階層設計する際などの設計情報の管理単位であり、ブロックの外部仕様となる情報を持つ。モジュールの実際の中身は、インターフェース設計情報や機能ブロック設計情報あるいは詳細なブロック図である。

【0044】[2] プロパティとテンプレート
見積もり評価用のパラメタの入力は、ユーザの任意のものとすると見積もりの基準が曖昧になるし、集計するシステムも設計者の数だけ必要になってしまう。見積もり評価の標準化と、設計者への情報入力を促すために、本システムでは、G U I (Graphical User Interface) によるパラメタ入力フィールドを設ける。このために、入力の雛形(テンプレート)を外部の定義ファイルとしてパラメタテンプレート保持部5に持つ構造とする。

【0045】システム起動の際に、パラメタ設定部22は、パラメタテンプレート保持部5からテンプレートファイルを読み込み、定義してある情報を解析し、定義されているパラメタの入力用のG U I ウィンドウを動的に作成する。これにより、ユーザである設計者がブロックやインターフェースにパラメタを設定しようとした際に、必要となるパラメタの入力を促す。パラメタのテンプレートには、必須項目、任意項目の区分を付けることにより、必要な情報を漏れなく設定させることができ、設計者個人による見積もり情報のバラツキを押さえることができる。

【0046】このテンプレートのファイルを設計対象の種類別などで用意することにより、装置に応じた見積もりパラメタを漏れなく入力することが可能になる。また、このテンプレートによって、後述する蓄積した設計資産の再利用のためにも、資産ブロックごとにブロック固有のパラメタの入力促進および入力が行えるようになっている。

【0047】[3] ライブラリブロックへのテンプレート登録

機能ブロックは、再利用を考えて設計し、その設計した資産を再利用するためにライブラリとして登録しておく場合が多い。このため、本システムでは、ブロック図入力・編集部2で得たブロック情報をライブラリ化してブロック情報記憶部6に記憶する。ブロック情報をライブラリとして登録する場合に、そのブロック固有のパラメタテンプレートを設定しておくことにより、そのブロックを使用する際にブロック固有の見積もりパラメタの入力促進を行うことができる。この場合のパラメタ入力促進用にも前述のテンプレートを用いる。再利用するためライブラリ化するブロック情報の中に、このプロパティテンプレートを埋め込んでおく。

【0048】図5は、ライブラリにノウハウとして蓄積されるブロック情報の例を示す。ブロック情報には、ブロックを示すシンボル、そのブロック固有の見積もり評

価ルール、プロパティテンプレート等が含まれる。

【0049】ブロック図面編集部21は、あるブロックを引用する際には、ブロック情報記憶部6のライブラリから該当するブロック情報のテンプレート情報を読み込み、そのテンプレート情報から入力促進G U I を動的に作成し、入力を促すようにする。

【0050】[4] ライブラリブロックへの見積もりルール登録

本システムでは、ライブラリブロックに、そのブロック固有の見積もり評価用のルールを登録することができる。見積もり評価ルールは、前述したパラメタテンプレートから作成された入力用のG U I に沿って入力されたパラメタを使って、各種の設計情報の見積もりを行うための簡単なプログラムである。

【0051】見積もり評価処理部3は、ブロック固有の見積もり評価ルールを用いて、そのブロックの見積もり計算を行う。設計システム全体の見積もり評価を行う場合には、見積もり評価ルールを持つ個々のブロックごとに、パラメタにより制御された見積もり評価ルールによる計算を行う。これらの計算結果を、システム全体で積み上げていく。

【0052】設計対象毎に評価予測の精度を上げるためにには、ノウハウの蓄積が必要であり、設計資産ブロックやマクロブロックなどに対して、それぞれ固有のパラメタ、算出方法を与える。

【0053】具体的には、ライブラリの機能ブロックに、その機能ブロックに与える固有パラメタのテンプレート、固有パラメタおよび設定対象全体に付与するパラメタから評価値を導く式、を定義可能とする。

【0054】ブロック図の編集時には、ブロック固有なパラメタテンプレート情報によりG U I ウィンドウを表示してパラメタの設定を促し、設計情報記憶部7内にパラメタ情報を格納する。

【0055】見積もり評価を行う際には、ユーザに設定されたパラメタと、予め定義されている見積もり評価ルールに基づいて規模、パワー予測などの計算を行う。図6は、評価に必要な汎用パラメタ値の見積もり評価ルールの例を示す図である。見積もり評価ルールは、計算方法を記述したプログラムである。基本的には、条件(i)と、条件を満たす場合または満たさない場合の演算の手続きからなる。

【0056】この見積もり評価ルールとパラメタとにより、ライブラリ化するブロック情報に設計ノウハウを蓄積することができる。見積もり評価と最終的な設計結果から、見積もり評価ルールや指定パラメタの更新を行うことで、将来のための見積もり評価の精度を上げていくことができる。

【0057】[5] 見積もり評価処理

見積もり評価の処理は、図7に示すように、ユーザインタフェース部4と、見積もり評価処理部3のブロックル

ール演算部31と全体見積もり演算部32とにより行う。

【0058】ブロックルール演算部31は、前出の[3]および[4]の項で述べたブロック固有のパラメタと見積もり評価ルールを使って評価を行う。全体見積もり演算部32は、各ブロックに指定されたパラメタ値、およびブロックルール演算部31で計算された値を使って設定対象全体の見積もり積算を行う。

【0059】見積もり評価処理の実行は、設計対象全体の見積もり積算の処理の中で、各ブロックの階層ツリーをり、ブロック個々の見積もり評価計算を行っていく。

[6] 階層設計とパラメタ設定

設計の詳細化が進むにつれ、個々のブロックの中を、さらにブロック図表記していく。その際に下位階層のブロックにもパラメタを指定していくことができる。これにより、より細かいパラメタ指示が行え、見積もり評価の精度を上げることができる。

【0060】上位階層のブロックおよびその下位階層のブロックの両方にパラメタが指定された場合には、下位階層のデータを優先するようとする。下位階層でパラメタが満たされない場合、すなわち特定のパラメタが指定されていないブロックがあった場合には、そのパラメタについては上位階層のブロックに付与されたものを採用する。

【0061】図8は、階層設計とパラメタ精度の関係を説明する図である。図8において、上位階層のパラメタ値PRM1は下位階層パラメタ値を積算した値である。

【0062】上位階層ブロックBLK-Aは、ブロックBLK-XX、BLK-D、BLK-Eの3個の下位階層ブロックからなる。上位階層ブロックBLK-Aのパラメタ値PRM1は500であるが、その下位階層ブロックの各パラメタ値が、ブロックBLK-XXではPRM1=110、BLK-DではPRM1=180、BLK-EではPRM1=180であったとすると、この場合、BLK-Aのより詳細化したレベルのパラメタの積算値が470となることから、BLK-AのPRM1の値“500”ではなく、下位の階層のパラメタの積算値“470”を使用する。

【0063】一方、上位階層のブロックBLK-CのPRM1の値が200であって、その下位階層のブロックのうちBLK-WWのPRM1が未設定である場合には、詳細化した下位階層のブロックのパラメタの積算値は使用せず、上位階層のブロックBLK-CのPRM1の値を使用する。

【0064】図8に示すPRM1というパラメタは、下位階層のパラメタ値を積算すれば、上位ブロックのパラメタになるという一例である。このような見積もり用パラメタには、例えば面積、コスト、消費電力などがある。当然、単純な積算では済まないパラメタもあり、例

えば遅延時間（サイクルタイム）や、信頼性パラメタなどのパラメタは、それぞれのパラメタに応じた見積もり評価ルールを持つ。なお、図8の例では、ブロックへのパラメタ設定だけであるが、前述したようにブロック間をぐインターフェースへのパラメタもある。具体的には、インターフェースへのパラメタとして、物理的な線の本数や、信号のトグル周波数、信号レベルやデバイスインターフェージなどがある。

【0065】また、上位階層と下位階層でのブロックのパラメタ値に明らかな矛盾がある場合、例えば回路規模を算出するパラメタが、上位のブロックよりもそれに含まれる下位のブロックの方が著しく大きい場合などは正常ではない。見積もり評価処理部3は、このような場合にパラメタ値の矛盾を検出する機能を持つ。

【0066】図9は、パラメタ矛盾検出処理を説明する図である。図9(A)は処理をするブロックの階層構造の例を示し、図9(B)は各ブロックのパラメタ情報のデータ構造を示している。

【0067】パラメタ矛盾を検出する仕組みは、図9(A)に示すようなデータ構造で、階層のブロックリンク情報と、図9(B)に示すようなパラメタを持つことより実現することが可能である。

【0068】各ブロックは、図9(B)に示すように、ブロック名、下位ブロックリンクの他に、パラメタ数分のパラメタ情報を持ち、各パラメタ情報には、自己のパラメタ値、下位ブロックに基づくパラメタ値、フラグ等がある。パラメタ毎にあるフラグの種類と意味は次の通りで、全体の積算を行う処理で各ブロックのデータを取り出すときの判断を行う。

【0069】・下位ブロックパラメタ算出済(下位パラメタ値を使用)

- ・下位ブロックパラメタ不完全(自己パラメタ値を使用)

- ・下位ブロックパラメタ未確認(下位ブロックを調査)

- ・下位ブロックなし

【7】カット・アンド・トライ評価と設計情報の保存
一般に、システムの検討時には、複数の実現手法で装置構成を検討する場合が多い。それぞれの装置構成や物理的な実現方法で見積もり評価を行い、各方法でのデータと見積もり評価結果を蓄積する。複数の評価結果を蓄積後、それらを比較検討するためのグラフ化などを行うGUIにより、最適解を探す支援を行うようにする。複数の評価結果の中での最適解を見定めたら、その時点の設計情報を呼び出すことができ、さらにその設計情報から構成方法を変更したり、評価パラメタをチューニングすることにより最適な実現方法を絞り込んでいく。

【0070】本システムでは、設計情報に対して細かい版数管理を行い、見積もり評価処理部3により見積もり評価を行う都度、設計情報の版数を上げて複数のデータを持てるようにする。この設計情報の中に評価結果データ

タも合わせて持つ構造とし、設計情報と評価結果とを合わせて管理する。

【0071】また、評価結果から設計情報をさらにアップデートする場合には、設計情報の版数が枝分かれしていくように管理する。このような版数管理の方法は従来からある技術であるが、設計情報と評価結果とを合わせて保持し、評価結果を比較検討できるところに特徴がある。

【0072】本システムで取り扱う設計情報は、設計の上流工程での設計データであるため、データ量は回路図などに比べるとはるかに少ないが、それでも設計情報全体を見積もり評価の度に保管するとデータ量が膨大になってしまう。

【0073】そこで、設計情報の全体を丸ごと保管するのではなく、設計情報は変更があったものだけ階層ブロック単位に版数を付けて保管する。そして、全体を管理するインデックス（どのブロックが何版であるかを管理するデータ）の版数を上げ、評価結果と共に設計情報記憶部7に保管する。見積もり評価結果のデータは、全体管理のデータベース（DB）に格納してもよく、また、別のDBやファイルに格納して、ポインタ情報のみ管理DB（この場合、設計情報記憶部7）に格納するようにしてもよい。

【0074】見積もり評価の実行タイミングは、そのシステム全体を構成する各ブロックの版数をチェックし、以前の評価時から変更があったブロックについて、版数を上げて保管する。そして、インデックスを作成し、見積もり評価データと共に保管する。各ブロックの版数は枝版数を持たず、単純にインクリメントされる。この管理方法により設計データの保管量の削減を図ることができる。

【0075】図10は、評価データの蓄積と設計版数の関係を示す図である。図10においては、見積もり評価データは別納して、ポインタ情報のみを設計情報記憶部7に格納する場合を例にしている。

【0076】図10（A）は、全体の版数2から枝分かれした版数2.1と、版数2.1におけるBLK-A、BLK-B、BLK-Cそれぞれの版数の関係を示している。全体の版数2.1では、各ブロックの設計情報の版数は、BLK-Aが第2版、BLK-Bが第2版、BLK-Cが第4版となっている。

【0077】図10（B）は、全体の版数管理のインデックスの例を示している。インデックスには、見積もり評価データへのポインタ情報と、各ブロックの版数情報を格納する。

【0078】本システムは、一例としてワークステーションやパソコンなどのコンピュータのハードウェアおよびソフトウェアによって実現される。図11は、本発明を実施する場合の装置構成の例を示す。本システムを実現するプログラムや設計データは通常は磁気ディスク1

03に格納され、プログラムを実行する際にメモリ102に読み込まれる。CPU101は、メモリ102に読み込まれたプログラムの命令をフェッチし実行する。

【0079】ブロック図やパラメタ情報も編集中はメモリ102上に置かれ、ディスプレイ104への表示やキーボードその他の入力デバイス105からの入力による編集処理がプログラムにより行われる。設計データは、編集を終えると磁気ディスク103に格納される。また、見積もり評価を行った結果も磁気ディスク103に格納される。

【0080】外部I/F106は他のワークステーションやパソコンなどと接続する装置で、本発明には必須ではないが、複数のワークステーションやパソコンを使って設計作業を行う場合の連携に使用する。

【0081】次に、LSIの設計を例に本発明の処理の流れを説明する。

（1）概略ブロック図入力処理

ブロック図入力・編集部2により、概略のブロック図を入力する。

【0082】ブロック図の入力では、ライブラリを用いずに任意にシンボルやシグナルを作成することもできる。任意のタイプのシグナル（ライン）を入力してシンボルを接続するとピンができる。なお、ブロック情報記憶部6に記憶したシステムマクロ等の資産ライブラリも入力に使用することができる。System、BWB、PCB、LSIなどの物理階層を気にしないで入力できる。

【0083】図12は、ブロック図の入力処理により入力された概略ブロック図の例を示す図である。

（2）パラメタの設定

ブロック図入力・編集部2により、図12に示すブロック図上で見積もり評価のためのパラメタを入力・設定する。パラメタテンプレート保持部5に記憶されたテンプレート定義ファイル等から自動作成したパラメタ入力画面を表示する。ユーザは、その項目に基づいてパラメタを入力する。

【0084】図13はパラメタ入力の例を示す図である。ブロック図のシンボル#2と#6とを結ぶシグナルのシグナルパラメタ入力ウインドウ110は、パラメタ

40 テンプレート保持部5に記憶されたテンプレート定義ファイルから自動作成される画面である。また、シンボル#6のモジュールパラメタ入力ウインドウ111も、同様にパラメタテンプレート保持部5に記憶されたテンプレート定義ファイルから自動作成される。シンボル#3のマクロパラメタ入力ウインドウ112は、ライブラリブロック固有のパラメタの入力画面である。ライブラリ毎にテンプレートを持っている。

【0085】ユーザ（設計者）は、これらの入力ウインドウ110、111、112等からパラメタを入力・変更し、任意にパラメタを設定する。例えば、シンボル#

6の任意に作成した論理モジュールには、フリップフロップ(F F)数やGate数、動作周波数、動作率などの汎用パラメタを設定する。シンボル#3のライブラリマクロに対しては、マクロ固有のパラメタ、実現方式や機能、Bitなどを設定する。シンボル#2～#6間のシグナルには、インプリメント時の信号本数(高速、通常)や周波数などを設定する。ROM、RAMには、Bit、Word、Columnなどを設定する。

【0086】図14は、各ブロックに付与されたパラメタ情報の例を示している。上記の入力ウインドウから入力されたパラメタ情報は、例えばシステム内部では図14に示すようなパラメタリストで管理される。

【0087】(3) 見積もり評価の条件の設定
統いて、見積もり評価の条件を設定する。図15は、見積もりの評価条件の設定の例を示す。ブロック図入力・編集部2により、図15に示すような見積もり評価条件設定ウインドウ120を、テンプレート定義ファイルから自動生成して表示する。このウインドウ120の入力画面から、インプリメントするテクノロジ(LSIシリーズ)やパッケージタイプ、スキャンタイプ、平均動作周波数、電源電圧等の設計条件を設定する。テクノロジ等を選択することで、そのテクノロジで指定可能なメニュー情報がパラメタテンプレート保持部5等から読み出され、メニューから選択する形で各種の条件を設定することができる。

【0088】図16は、図15に示すような評価条件設定ウインドウ120からの入力によって設定された見積もり評価条件の具体例を示している。

(4) 見積もり評価処理の実行

設定された見積もり評価条件に基づき、見積もり評価処理部3は、見積もり評価を行う。ブロックが階層構造であれば、各ブロックの見積もり評価処理と下位の階層ブロックの評価結果を積算した全体の見積もり評価とを算出する。

【0089】さらに、カット・アンド・トライ等の物理分割の見積もり評価を行うことができる。図17は、物理分割の見積もり評価結果を表示する画面の例を示す。例えば、チップ分割の場合のように、物理的に分割する範囲を指定することで、その分割の範囲ごとに、ピン数、合計BC数、バルクに対する使用率、消費電力等の見積もり評価結果を確認することができる。図17において点線および一点鎖線の枠は、分割の範囲を表している。他に、色を変えて表示するというようなこともできる。各分割の範囲ごとに、評価結果のレポートが表示される。

【0090】分割範囲の変更、見積もり評価の実行、見積もり評価結果の表示を繰り返すことで、カット・アンド・トライを容易に実行することができる。

(5) 物理分割(階層構造の再構築)

見積もり評価処理部3により、カット・アンド・トライ

等を行い、物理的な見積もり評価を行った後、物理階層への切り出し(例えばチップ分割)を行う。

【0091】図18は、評価結果から物理階層を決め、ブロック図の階層を変更した例を示す。見積もり評価結果に基づき、図18(A)に示すような分割範囲で物理階層を定め、図18(B)に示すようにブロック図の階層(レベルレイヤ)を変更することができる。このような物理階層への切り出しは、ブロック図入力・編集部2等が持つシステム記述エディタの階層組み替え機能を用いて容易に行うことができる。

【0092】(6) 詳細設計

詳細設計では、各モジュールの中を、さらにブロック図による構造設計とHDLにより動作・機能設計とによって詳細化していく。

【0093】図19は、HDL記述による詳細化の例を示す図である。図19に示すように、詳細化によって、シンボルからHDL記述等への詳細情報へのリンクがなされる。このブロック図の構造記述部は、HDLネットリスト生成を行う。動作・機能設計を行うHDL記述については、BL、RTLを問わない。

【0094】図20(A)は、見積もり評価条件に基づく各ブロックの評価結果の出力例を示している。図20(B)は、LSI全体の評価結果の出力例を示している。もちろんこれらは一例であり、任意の形式で評価結果を出力することは、以上の説明から容易に実施することができる。

【0095】<本システムによるエディタ機能>図21に、本発明のシステムをより発展させたエディタの内部構成を示す。そして、本エディタは、環境設定ファイル、ライブラリ、データベース、プロパティ・テンプレート等からデータを引き出し編集するとともに、環境設定ファイルやデータベースにはデータを登録できるようになっている。そして、本エディタから評価ツールや、他ツールとして他のアプリケーションを起動しこれらをXウインドウやウインドウズなどのOS等で本エディタとともに連携して処理することができる。なお、本実施例ではXウインドウ環境下のもとで、エディタ等のアプリケーションとXウインドウとの間の画像処理系APIとしてGALAXY API(米国VIX社製)を用いることを想定して説明している。

【0096】以下、図21に示す構成を順次説明する。

(1) 環境設定ファイル

エディタの編集・表示機能など各種のモード設定、画面の配色、画面サイズなど動作、編集環境などユーザーが設定可能な項目値を定義したファイルである。

【0097】(2) 環境Load/Save手段

上記、環境設定ファイルをメモリテーブルに展開(Load)する機能と、環境設定処理部により変更されたメモリテーブルをファイルに書き出す(Save)機能を持つ手段である。

【0098】(3) 環境設定Table

前述の環境設定ファイルをメモリ上に展開したテーブル（C++のクラス定義）であり、各種の編集処理、画面表示処理に於いて、本テーブルを参照し、設定された動作を行う。また、後述の環境設定処理部によって設定内容の変更を行うことができる。

【0099】(4) 環境設定処理部

エディタの編集・表示機能強など各種のモード設定、画面の配色、画面サイズなどをユーザーが設定するためのGUI部である。

【0100】ノートブックタイプ・のメニューを使い、カテゴリ毎にページ（タブ）を分ける。設定ウィンドウを開き、設定項目毎にボタン、選択メニュー、文字列表示・入力などが行え、変更された内容を環境設定テーブルに反映する。

【0101】(5) ライブラリ（Library）/データベース（Database）

本システムではライブラリ、データベースの構造を同一としており、運用時のディレクトリで切り分けることとする。

【0102】ライブラリには既設計資産、マクロ、セルなど各レベルのモジュール（Module）が含まれ、データベースはそれらを組み合わせた、あるいは新規モジュールによるブロック図データが格納される。

【0103】従来の機能図、回路図にあるページの概念ではなく、1階層（モジュール）単位に扱う。データベースの形式はUNIXのファイルシステムそのもので、モジュール毎にファイルを分ける。細かくは、モジュール外皮と、中身であるブロック図本体のファイルに別れている。モジュール外皮とは、自己シンボル、ポート（port）情報、プロパティ（property）、テンプレート（template）などの情報を格納するテーブルである。モジュールは既存設計資産、マクロ、セルなどの各レベルの情報であり、自己シンボル形状、ポート情報、プロパティ情報、テンプレート情報を持つ。

【0104】ファイルの中身はASCIIファイルであり、シーケンシャル・アクセスしてメモリ上に読み込んで編集操作を行う。

(6) ライブラリ/データベース展開（Library/DB Load）

前述のライブラリ、データベースをメモリテーブルに展開する処理を行う部分で、モジュール外皮とブロック図本体の展開部から構成される。

【0105】モジュール外皮はデータテーブル（Data Table（モジュール））に展開され、ブロック図本体はデータテーブル（Data Table（ブロック図））上に展開される。

【0106】各ファイルはシーケンシャル・アクセスするが、必要な部分だけを高速に読み出すために部分的に展開する機能をもつ。指示されなかった部分は読み飛ば

し、メモリテーブルに展開しない。

【0107】これによりファイルのアクセス時間そのものは短縮されないが、ASCIIファイルを解析し展開する時間を省くことができる。読み込み時には”read”モード（参照だけ行う）、”Writeモード（編集を行い、書き込むことが前提の読み出し）を指定でき、同一モジュールに対しての排他・共有制御機能を持つ。

(7) データベース保存（DB Save）

10 メモリテーブル上で編集されたブロック図データ、モジュールデータをデータベースに書き込む処理を行う。 Library/DB Loadで述べた排他制御を行う。

(8) データテーブル（Data Table（ブロック図））

ブロック図を構成するシンボルやラインなどの要素を格納するテーブルで、本テーブル上でブロック図の編集処理を行う。

【0108】既存データの場合、DB Load処理によりデータベースから本テーブルが作成される。新規設計データの場合には編集機能により本テーブルにシンボルやラインが追加、削除などが行われ、ブロック図データが構築されていく。

(9) データテーブル（Data Table（モジュール））

編集中のブロック図に対応するモジュール外皮情報であり、自己シンボル、ポート（port）情報、プロパティ（property）、テンプレート（template）などの情報を格納するテーブルで、本テーブル上でモジュールの編集処理を行う。

30 【0109】既存モジュールの場合、DB Load処理によって本テーブルが作成され、新規の場合にはモジュールの編集処理によって作成される。ライブラリ引用時のライブラリシンボル形状は本テーブルではなく、DataTable（ブロック図）の方に格納される。本テーブルは設計データであるモジュールの情報のみ扱う。

(10) プロパティ テンプレート（Property Template）ファイル

ブロックシンボルやインターフェース（接続定義）に付与可能なパラメタ（プロパティ：Property情報）を外部定義したファイルである。これにより、プログラムの変更が最小限でプロパティ入力用のパネルを作成可能である。これはライブラリブロックにも定義可能で、同一の記述方法をとる。この場合、ACCIファイル形式で一般的なテキストエディタを使用して定義する。

(11) テンプレート展開（Template Load）処理

前述のProperty Templateのファイルを読み込んでメモリ上のテーブルに展開する処理である。

(12) テンプレート テーブル (Template Table)

ブロックシンボルやインタフェースに対する汎用的なプロパティ情報のテンプレートとなるメモリ上のテーブルであり、ここの情報を元にGUIのダイアログボックスを生成し、ユーザーにパラメタ値の設定を促す。

(13) ブロック図表示処理

データテーブル (Data Table) に格納されているシンボル、ラインなどの要素を編集ウィンドウに表示（消去を含む）する処理を行う。

【0110】表示に当たっては、ズーム、パン、画面サイズ変更などを処理し、画面の表示範囲内のみを表示するクリッピング処理を行う。表示する項目（ブロック名称、固有名、信号名、プロパティ値など）について表示／非表示の制御を行える。

(14) 基本描画機能

ブロック図や階層シンボルを編集画面に表示（消去を含む）するための基本的な图形描画ルーチンであり、直線、連続線、円、円弧、矩形、文字などの描画、消去を行う。

【0111】表示图形の重なりについては、重なった部分の色が変わり、一方の图形を消去した場合には他方の图形（の色）が元に戻る。图形のドラッグやラバーバンド機能、マウスポイント图形の変更についても取り扱う。

(15) ブロック図編集機能

シンボルや信号線などブロック図を構成する要素の追加、削除、移動、複写などの編集機能であり、エディタのメインとなる部分である。

【0112】範囲を指定しての削除、移動、複写や、信号線が追従するシンボル移動など機能が多い。また、下位階層の上位階層への展開機能や、その逆の1階層下のモジュール生成などが可能である。

【0113】エントリのスタイルを固定せず、ユーザーフレンドリで自由なオペレーションでブロック図を描ける。

(16) ブロック／シンボル (Block / Symbol) 編集コマンド処理

編集画面上で行われるマウスアクションを取り出し、各種の編集機能に振り分ける処理を行う。

(17) モジュール編集機能

モジュールのシンボル（階層シンボル）情報、プロパティテンプレート、計算式ルールなどを編集する機能である。

【0114】ブロック図やVHDL言語記述が存在する場合にはポート (port) 情報からシンボルを自動生成する機能を備える。

(18) Property 設定パネル生成、設定ブロックシンボル、モジュール、ライン、インタフェースに設定するプロパティ情報の入力用ダイアログボック

スをダイナミックに生成する機能であり、設定された内容をデータテーブルに格納する機能を備える。

【0115】プロパティは複数の要素に対して与えられるが、全てこの共通機能を使用して設定する。

(19) フレーム、メニュー、ボタン (Frame, Menu, Button) 作成

エディタ全体のGUIフレームとなるFrame, Menu, Buttonなど各種のウィジェット、コールバックルーチンなどを定義する部分である。

10 【0116】この部分はGUIビルダーや市販のウィジェットキットが有効に活用できる。

(20) 他ツール起動

他のアプリケーションツールの起動、パラメタなどを渡す処理を行う。

【0117】これは、ブロック図中のシンボルの中身がVHDLの場合や、他のツールで作られている場合にテキスト編集ツールや他のアプリケーションを起動して、その編集や参照が行えるようにするためである。

(21) 評価ツール連携

20 物理分割などの支援を行う評価ツールとの連携を行う部分である。

【0118】設計データの受け渡し（当初はDBインターフェース）や分割範囲などの情報の受け渡しや評価ツールの起動、評価結果の画面へのフィードバック処理を行う。画面へのフィードバックとしては、ブロック図中のシンボルへのゲート規模や電力表示、色の変化、リスト表示、グラフ化などがある。以上が、図21の各構成の機能であるが、次に、この図21に示した構成によりブロック図を作成する場合の特徴を説明する。

30 【0119】本発明の特徴は、個々のブロックとブロック間を接続する線（以下、インターフェースと呼ぶ）を同レベルで扱い、共に階層的な表現ができるようにしている点である。

【0120】従来のブロック図や回路図の入力では、ブロックが機能を表現し、接続するラインは単なるワイヤーであった。本発明のブロック図ではラインが単純なワイヤーではなく機能を持たせるとができるようになる。さらに、ラインへの物理的なパラメタも付与可能としている（図22参照）。

40 【0121】具体的にラインの機能とは、プロトコルそのものであったり、プロトコルを実現する機能などである。物理的なパラメタとはそのプロトコルを実現した場合の実際のワイヤー本数や機能を具体化するための回路規模、また信号のトグル周波数であったりする。以上のように設計情報を属性プロパティとして管理する。ブロックには階層設計の下位ブロックや他の環境で設計した設計データ、ドキュメントデータなどへのリンク、見積もり評価用のパラメタを持ち、インタフェースにも同様に設計データへのリンク情報、評価パラメタなどを持つ。

【0122】図22は、図2とほぼ同様であるが、ドキュメント情報をプロパティとしてリンクしている点が、図2と異なる。機能ブロック、インターフェースそれぞれ抽象レベルの異なる設計情報を抽象レベルレイヤー（アーキテクチャ）と云う概念で複数持つことができる構造とする。これは図3に示すように、ブロック図で表現した機能ブロック、インターフェースのシンボルに対して抽象的な記述から具体的な詳細記述まで複数レベルの設計情報にリンク可能な構造である。

【0123】これにより抽象度の高いレベルから設計をはじめ、段階的に詳細化する設計手法がとれるようになる。以上を管理し、設計情報を保持するために、図4に示すようなデータ構造を持つ。図中、プロパティの箱が設計情報へのリンクや、見積もり評価用のパラメタなど複数のパラメタを保持し、設計データ全体のリンク構造を作っている。

【0124】ここでモジュールとは階層設計する際などの設計データの管理単位であり、ブロックの外部仕様となる情報を持つ。実際の中身はインターフェース設計情報や機能ブロック設計情報、あるいは詳細なブロック図である。

<設計データのリンクの方式・手順>

(1) 図23に設計データのリンク方式を示す

まず、設計データをリンクしたいオブジェクト（シンボル・ライン等）をクリックする。

すると、属性設定パネルが表示されるので新規の場合はアーキテクチャ名を入れてリターンキーを押す。

次いで、新しいTABが作成されて新規パネルが表示される。その場合はそのパネルにリンクするデータ名を入力または選択する。

既存の場合はリンクしたいTABを選択してパネルを全面に表示させそのパネルにリンクするデータ名を入力または選択する。

<外部プロパティとテンプレート>評価用のパラメタ入力は、設計者への情報入力を促すために、GUI (Graphical User Interface) によるパラメタ入力フィールドを設けている。本システムでは評価情報の変更にフレキシブルに対応するために入力の雛形（テンプレート）を外部の定義ファイルに持つ構造とする。

【0125】システム起動の際にテンプレートファイルを読み込み、定義してある情報を解析し、定義されているパラメタの入力用のGUIウインドウを動的に作成する。これにより、ユーザーである設計者がブロックやインターフェースにパラメタを設定しようとした際に、入力が必要なパラメタの入力を促す。パラメタの雛形には必須項目、任意項目の区分を付けることにより、必要な情報を漏れなく設定させることができ、設計者個人による見積もり情報のバラツキを押さえることができる。

【0126】このテンプレートのファイルを設計対象の

種類別などで用意することにより、装置に応じた見積もりパラメタを漏れなく入力することが可能になる。以下に外部テンプレートの構成を示す。

(1) 設計データと外部テンプレートの関係

図24に設計データと外部テンプレートの関係を示す。

【0127】本システムが起動すると同時に外部テンプレートファイルが読み込まれ、各設計データが設計者に指定されると同時に対応する情報がパネルに表示される。

10 (2) 処理フロー

図25に処理フローを示す。

【0128】ここで、まずシステムを起動する（ステップ101）。次いで、外部プロパティテンプレートファイルからファイル読み込みをする（ステップ102）。その後、プロパティファイルの解析を行い（ステップ103）、画面レイアウトを作成した後（ステップ104）、プロパティ設定用パネルを出力する（ステップ105）。

(3) 外部テンプレート（定義例）

20 図26に定義例を示す。

(4) 外部テンプレート（画面例）

図27に出力画面例を示す。

<シンボル編集機能>次いで、ライブラリレスシンボルの編集機能を説明する。

【0129】従来エントリでは図28で示すようにシンボルを入力する前に、入力するシンボルを作成することが必要であった。本システムでは、図29～図31に示したように、ブロック図編集において任意の大きさのシンボルを作成し、そのシンボルに対してライン（線）を任意の場所へ繋ぐだけで自動的にPORTを作成する。

30 【0130】以下に入力の手順を示す。

ツールバー又はメニューからシンボル作成機能を選び、テンプレートメニュー（矩形、台形、ALUタイプなど）を表示。このウィンドウはクローズするまで表示される。

テンプレートメニューから所望の形状をクリックして選ぶ。

配置したい位置にマウスポインタを置き、クリック（押すだけ）し、対角線方向にドラッグするとシンボル

40 形状のドラッグが伸びていく。

マウスボタンを離すとサイズが確定し、シンボル形状ができる。

【0131】対角線のドラッグ方向は左下→右上、右上→左下、右下→左上、左上→右下のいづれでもよい。

(1) シンボル入力フローを図30に示す。

(2) さらに、ラインの入力方法について図31に示す。

【0132】ツールバー又はメニューからライン入力機能を選ぶ。

50 始点をクリック（押すだけ）するとラバーバンドが始

まる。

終点位置にマウスポインタを移動し、マウスを離すと、入力完了。

【0133】 続いてクリックする際に折れ曲がり点ができる。

矢印の大きさ長さ、線の太さ、種類、色等はメニューで変更可能である。

ラインは自由な角度で入力できる。デフォルトでは水平、垂直のみとする。

【0134】 Shiftキーを押下げしながら入力すると自由な角度で描画可能である。

<階層間インターフェイス>次に、階層間インターフェースを図32に基づいて説明する。

【0135】シンボルの階層情報定義及び接続するシグナルと下位階層のportの接続方式は以下のような機能を持つ。シンボルと階層情報のリンクはシンボル属性定義パネルで行う。このパネルではアーキテクチャ別に複数の情報を一つのシンボルにすることができる。

【0136】更にそのシンボルに関連するドキュメントデータのリンクも定義できる。シンボルのportと下位階層のシグナル結合はシンボルPortコネクトパネルで行う。コネクトした下位シグナルの属性は全て上位のPortの属性に複写される。このため、下位の階層で設計したブロックの内容はそのまま、上位の階層へと反映され、上位の階層で設計したブロックの内容はそのまま下位の階層に反映される。

<階層シンボル自動作成>階層シンボル自動作成機能を図33に従い説明する。

【0137】VHDLのentity記述、またはモジュールに対応付けられたブロック図のport情報から、シンボルを自動作成する機能である。ファイル一覧からブロック図または階層シンボル編集ウィンドウにドラッグ&ドロップすることによって生成可能である。

【0138】以下に機能を示す。シンボルの左右に配置するピン数に合わせて形状の縦方向サイズを決める。port名、i/o種別、信号の型、Vectorのレンジ、portの並び順を情報とする。シンボル形状のタイプ、幅、i/oピンの位置、ピンの並び方などは、オプションのダイアログボックスから選択可能とする。項目は以下の通りである。

【0139】シンボル形状タイプ

- ・矩形、台形等

- ・形状の幅（段階的な選択で可）

ピンの配置

- ・input, output, inoutそれぞれ左右どちらに配置するかピンの並び順

- ・port名の昇順、降順、entityに記述されていた順又はブロック図の配置順

(1) 以下に構成と手順を示す。

【0140】図33のように、シンボル形状のタイ

プ、幅、i/oピンの配置、ピンの並び等をオプションパネルで設定しておく。

ファイル一覧から変換したいファイル名を選択または、マウスでドラッグしてシンボル編集ウィンドウまで移動する、シンボル編集ウィンドウでマウスを離すとえd登録しておいた方にしたがってシンボルの形状作成が開始される。

<階層展開、合成の機能>本発明では、機能分割に柔軟に対応するため複数の機能ブロックを指示し、ひとつの階層にしたり、その逆の操作をすることが可能である。

【0141】図34に構成と手順を示す。

(合成処理)

ブロック図上で合成したいシンボルを選択する（マウスで範囲指定、またはマウスクリックによる選択）。

【0142】選択したシンボル群で新たな階層シンボルを作成する。

port情報生成。

下位シンボル生成。

【0143】(展開処理)

展開したいシンボルを選択する（複数化）。

展開機能を起動する。

【0144】下位階層をロードして上位にブロック図に展開する。

シンボル（图形情報）を生成。

【0145】(展開処理) 展開はこの逆で、下位の階層の内容を上位階層に展開する。その際にはシグナル名、instance名の重複などに注意する。

【0146】シンボルへのプロパティの指定は形状確定の直後に設定するか、あとで設定するか切り替えられる。シンボルの領域がシグナルの端点と接する場合にはportの生成を行う。次に、図35に本CADシステムをワークステーションやパソコンなどのコンピュータ上のソフトウェアとして実現したときの構成を示す。また、ハードウェアで実現しても構わない。

【0147】図35で、システムを実現するプログラムや設計データは通常はディスクに格納され、プログラムを実行する際にメモリに読み込まれCPUを動作させる。ブロック図やパラメタ情報も編集中はメモリ上に置かれ、ディスプレイへの表示やキーボード等から入力された編集処理がプログラムにより行われる。設計データは編集を終えるとディスクに格納される。また、見積もり評価を行なった結果もディスクに格納される。

【0148】外部I/Fは他のワークステーションやパソコンなどと接続する装置で、本発明には必須ではないが、複数のワークステーションやパソコンを使って設計作業を行なう場合の連携に使用する。

【0149】

【発明の効果】以上説明したように、本発明によれば、装置設計の上流工程、すなわち設計の早期段階での高精度で実現性のある見積もりを行うことによって、最適ア

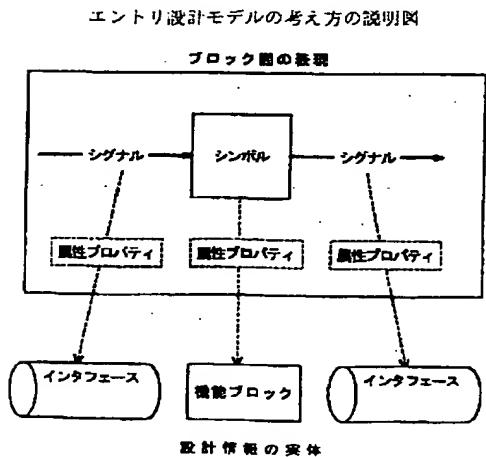
アーキテクチャや装置構成の実現を図ることができる。

【0150】また、設計や見積もりのノウハウをライブラリとして蓄積でき、さらに見積もり精度を高めていくことができる。さらに、アーキテクチャ検討段階でエントリした設計情報は、下流工程へ継承することが可能となり、下流工程での再エントリ工数や設計ミスを削減することが可能となる。

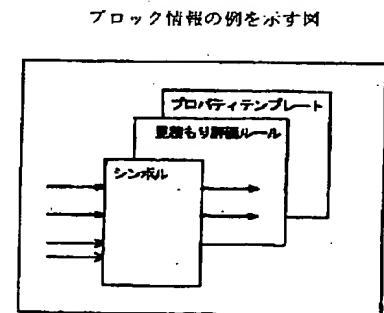
【図面の簡単な説明】

- 【図1】本発明のブロック構成例を示す図である。
- 【図2】エントリ設計モデルの考え方の説明図である。
- 【図3】設計情報のレベルレイヤの例を示す図である。
- 【図4】設計情報のデータ構造の例を示す図である。
- 【図5】プロック情報の例を示す図である。
- 【図6】見積もり評価ルールの例を示す図である。
- 【図7】見積もり評価の処理の説明図である。
- 【図8】階層設計とパラメタ精度の関係を説明する図である。
- 【図9】パラメタ値検出の処理を説明する図である。
- 【図10】見積もり評価データの蓄積と設計版数の関係を説明する図である。
- 【図11】本発明の装置構成の例を示す図である。
- 【図12】概略プロック図の例を示す図である。
- 【図13】パラメタ入力の例を示す図である。
- 【図14】各プロックに付与されたパラメタ情報の例を示す図である。
- 【図15】見積もり評価条件の設定の例を示す図である。
- 【図16】見積もり評価条件の具体例を示す図である。
- 【図17】物理分割の見積もり評価結果の例を示す図である。
- 【図18】物理階層への切り出しの例を示す図である。
- 【図19】HDLによる詳細化の例を示す図である。
- 【図20】見積もり評価結果の具体例を示す図である。
- 【図21】エディタ内部構成を示す図である。

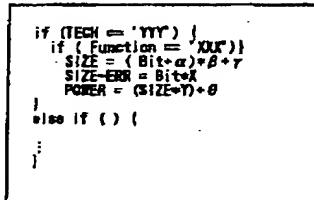
【図2】



【図5】



見積もり評価ルールの例を示す図



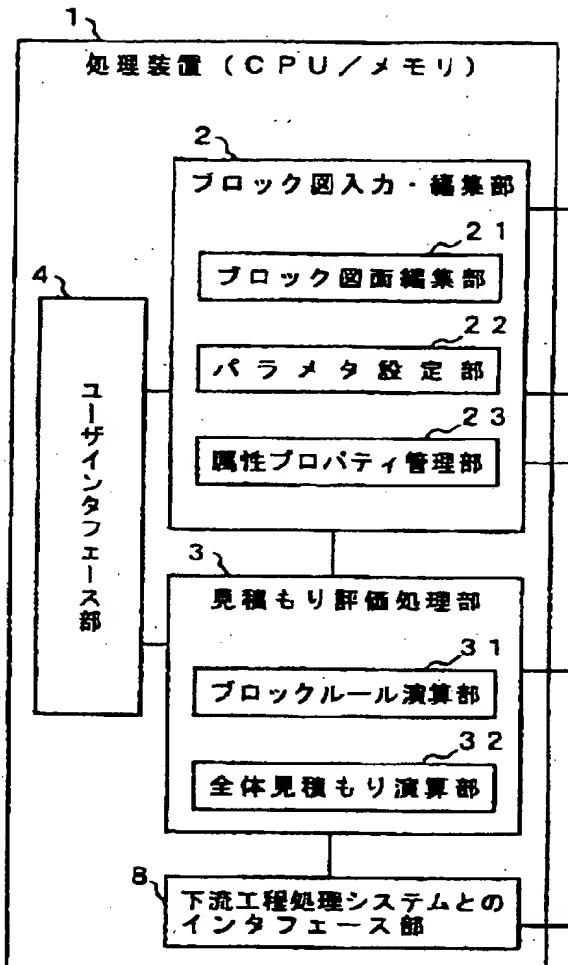
【図6】

【図22】エントリ設計モデルの考え方を示す図である。

- 【図23】設計データのリンク方式を示す図である。
 - 【図24】設計データと外部テンプレートの関係を示す図である。
 - 【図25】プロパティ設定の手順を示すフローチャートである。
 - 【図26】外部テンプレート定義例を示す図である。
 - 【図27】画面例を示す図である。
 - 【図28】従来型エントリを示す図である。
 - 【図29】本発明のシステムにおけるエントリを示す図である。
 - 【図30】シンボル入力フローを示す図である。
 - 【図31】ライン入力を示す図である。
 - 【図32】階層間インターフェイスを示す図である。
 - 【図33】階層シンボル自動生成を示す図である。
 - 【図34】階層展開・合成を示す図である。
 - 【図35】ソフトウェアで実施した場合の装置構成図を示す図である。
 - 【図36】プロック図の例を示す図である。
- 【符号の説明】
- 1 処理装置 (CPU/メモリ)
 - 2 ブロック図入力・編集部
 - 2.1 ブロック図面編集部
 - 2.2 パラメタ設定部
 - 2.3 属性プロパティ管理部
 - 3 見積もり評価処理部
 - 3.1 プロックルール演算部
 - 3.2 全体見積もり演算部
 - 30 4 ユーザインターフェース部
 - 5 パラメタテンプレート保持部
 - 6 ブロック情報記憶部
 - 7 設計情報記憶部
 - 8 下流工程処理システムとのインターフェース部

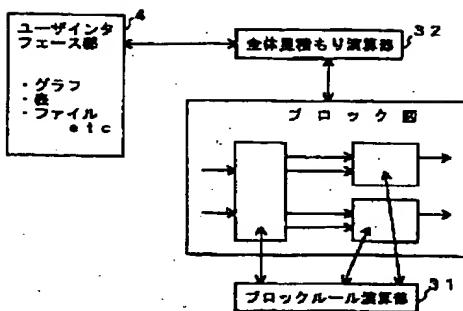
【図1】

本発明のブロック構成例を示す図



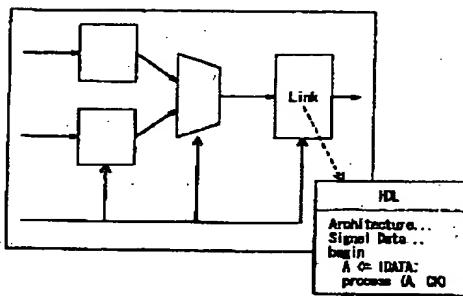
【図7】

見積もり評価の処理の説明図



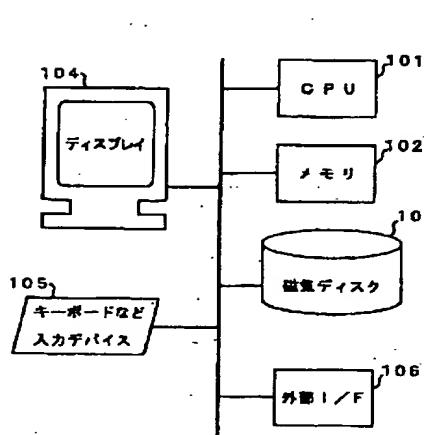
【図19】

HDLによる詳細化の例を示す図



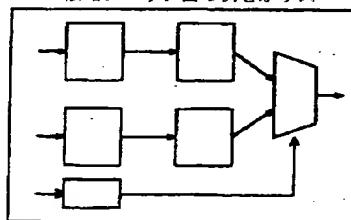
【図11】

本発明の装置構成の例を示す図



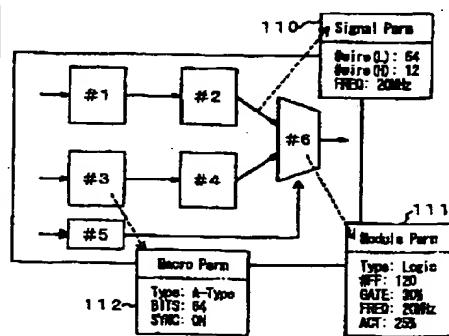
【図12】

概略ブロック図の例を示す図



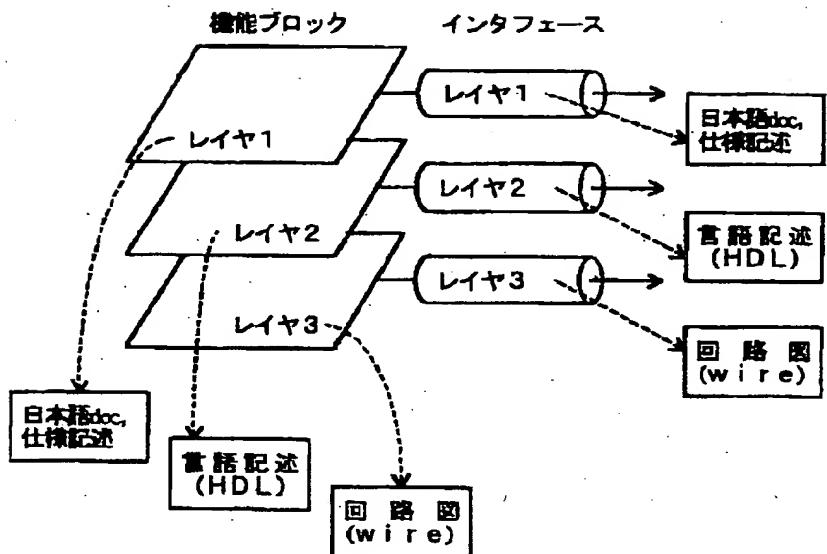
【図13】

パラメタ入力の例を示す図



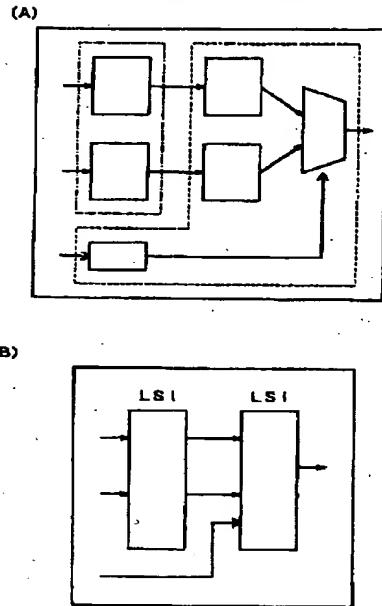
【図3】

設計情報のレベルレイヤの例を示す図



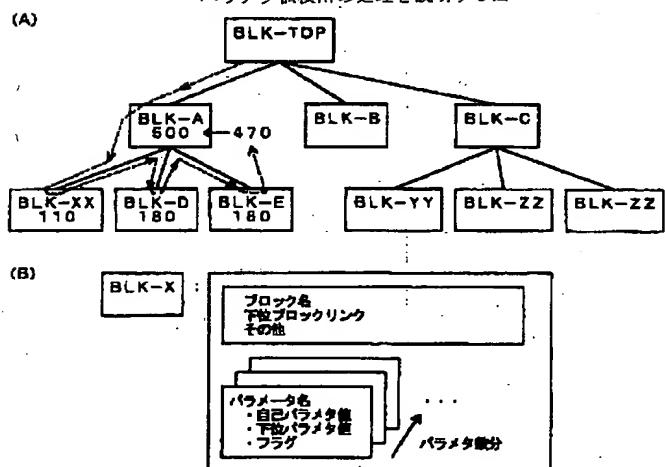
【図18】

物理階層への切り出しの例を示す図



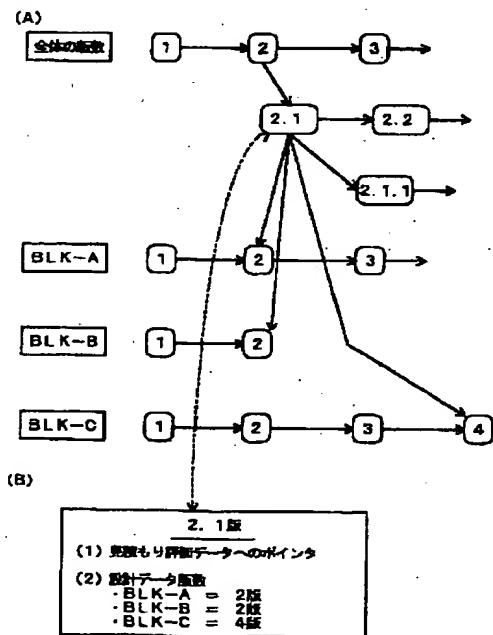
【図9】

パラメータ値検出の処理を説明する図



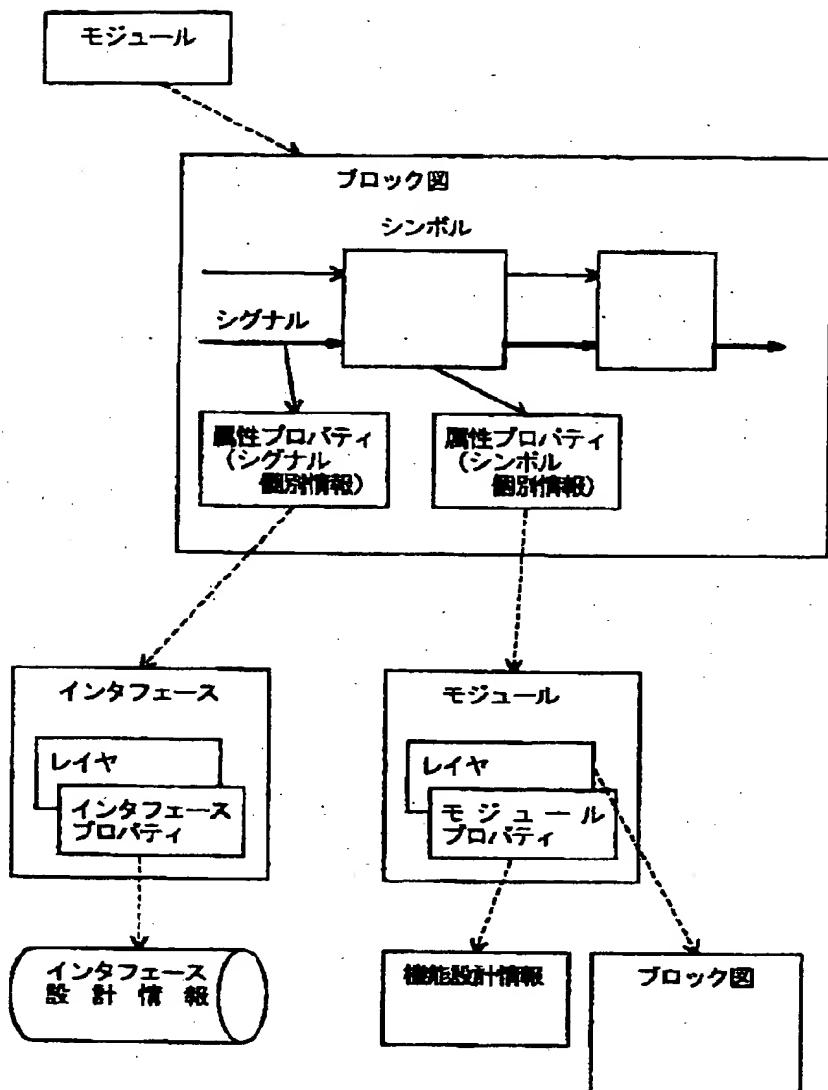
【図10】

見積もり計算データの蓄積と設計版数の関係を説明する図



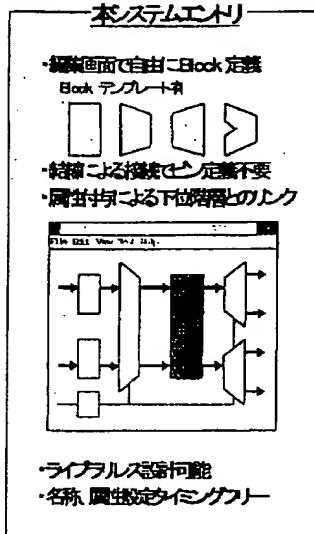
【図4】

設計情報のデータ構造の例を示す図



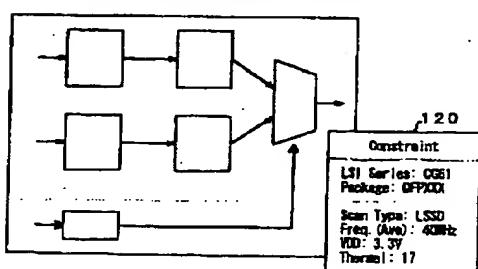
【図29】

本発明のシステムにおけるエントリを示す図



【図15】

見積もり評価条件の設定の例を示す図



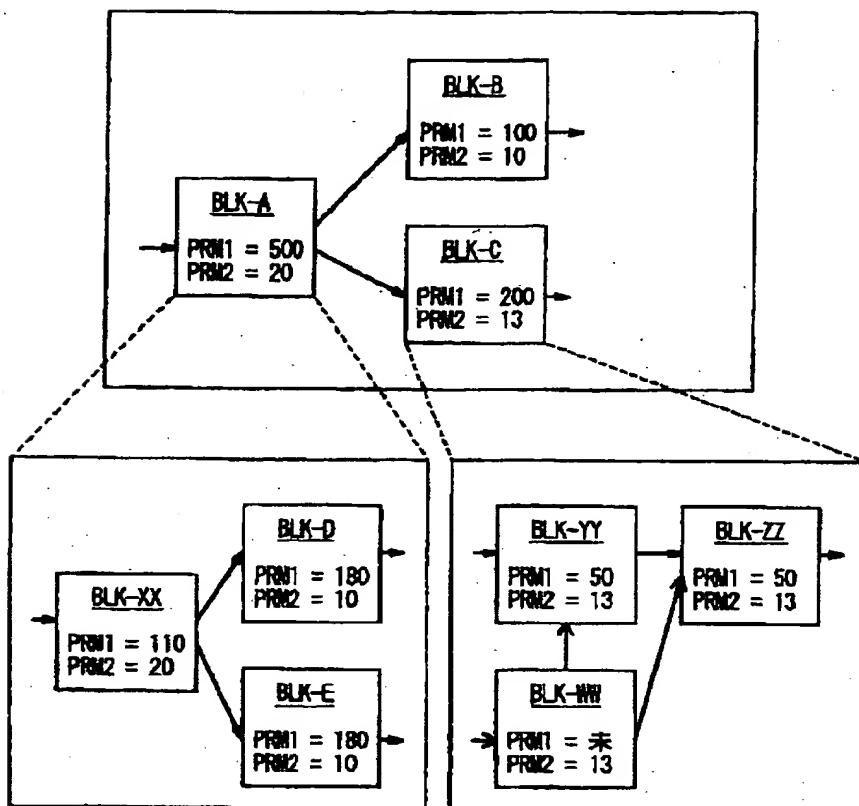
```
/* Estimation Condition
//EML:/home/ARCHICAD/TestDB/er1/CG51.ari Vendor:FJ Version:1 LastUpdate:7/25/1997 Comment:TestVersion
Property          Value
TEMP_AVERAGE_MAX 35
WIND              0m/s(multi)
VOLTAGE           3.3
SCAN_FT           Yes
POWER_TYPE        single
FREQUENCY_MHz    40
PACKAGE           PFP-64P-40M
SERIES            CG51P10
/* End of Estimation Condition
```

【図16】

見積もり評価条件の具体例を示す図

【図8】

階層設計とパラメタ精度の関係を説明する図



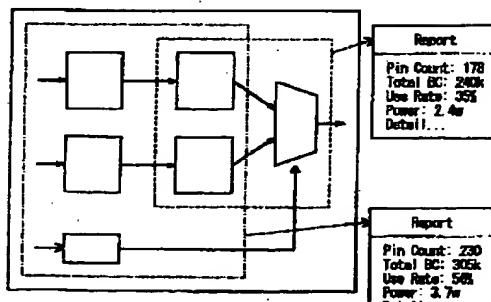
【図14】

各ブロックに付与されたパラメタ情報の例を示す図

```
/**Parameter List
//PropType means Used-property C:Common or D:Distinct
Module Architecture ModuleID PropType Property Value
C_MODULE SIMPLE_ARCHI CC01
  C MODULE_TYPE LOGIC
  C FT_NUM 300
  C GATE_RATE 15
  C GATE_DC 5000
  C GATE_AREA 12000
  C ACTIVE_RATE 30
  C MODULE_TYPE LOGIC
  C FT_NUM 100
  C GATE_RATE 20
  C GATE_DC 5000
  C GATE_AREA 8000
  C ACTIVE_RATE 20
  C MODULE_TYPE RAM
  C MEMORY_TYPE single
  C ATY 32
  C WORD 256
  C COLUMN 4
  C MODULE_TYPE LOGIC
  C CLEAR 0
  C LOAD 0
  C BIT 16
  C MODULE_TYPE RAM
  C FREQUENCY_MHz 40
  C ATY 44
  C WORD 1024
  C COLUMN 16
```

【図17】

物理分割の見積もり評価結果の例を示す図



【図20】

見積もり評価結果の具体例を示す図

(A) **Result by Module**

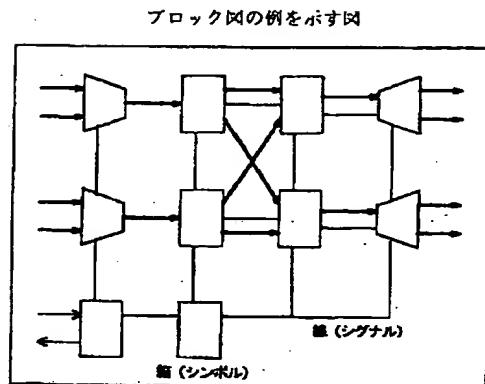
```
/*Module Data, $:Children SUM
Module      Architecture ModuleID Body Type Size(BC) Area ] PinNo POWER
C_MODULE    SIMPLE ARCHI AC01  BLOCK LSI  18561 (S) 22861 (S) - 0.13 (S)
D_MODULE    SIMPLE ARCHI CC03  VBE1  LSI  6000 (M)  9200 (M) - 0.09 (M)
C2_MODULE   SIMPLE ARCHI CC02  -  LSI  9061 (M)  9551 (M) - 0.04 (M)
AI_MACRO    SIMPLE ARCHI CC01  -  LSI  3500 (M)  4600 (M) - 0.00 (M)
B_MODULE    SIMPLE ARCHI AB01  -  LSI  8573 (M)  8573 (M) - 0.02 (M)
D_MODULE    SIMPLE ARCHI AB01  BLOCK LSI  9500 (S) 13800 (S) - 0.09 (S)
A_MODULE    SIMPLE ARCHI AB01  -  LSI  6000 (M)  9200 (M) - 0.09 (M)
AI_MACRO    SIMPLE ARCHI AB01  -  LSI  3500 (M)  4600 (M) - 0.01 (M)
/**End of Result by Module
```

(B) **Summary**

```
/*Result by System
Size(BC Area ) PinNo POWER rj
36634 45234 49 0.25 50

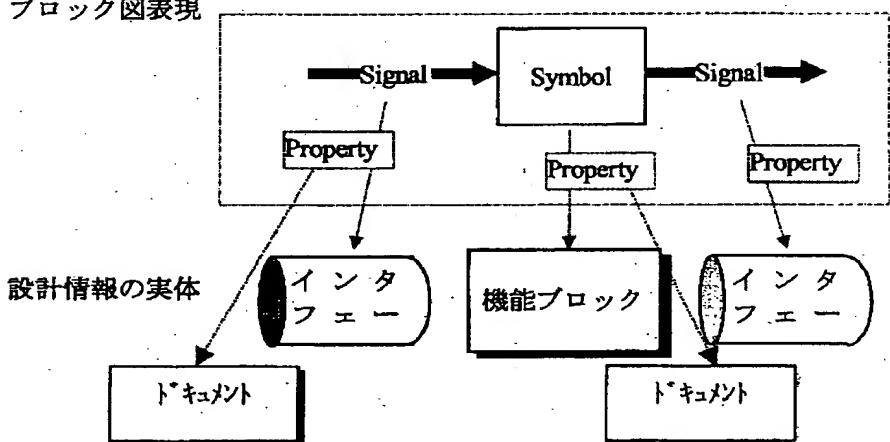
/*Chip Capacity by Lai Series
Series TotalBC UsedBC MaxPower Capacity:BC Area Pin#
CR61K71 1583808 1176258 506 3 3 9
CR61K59 1148840 849664 472 4 5 10
CR61K45 576281 576281 477 6 6 10
CR61K77 57750 57750 6 6
```

【図36】



【図22】

ブロック図表現

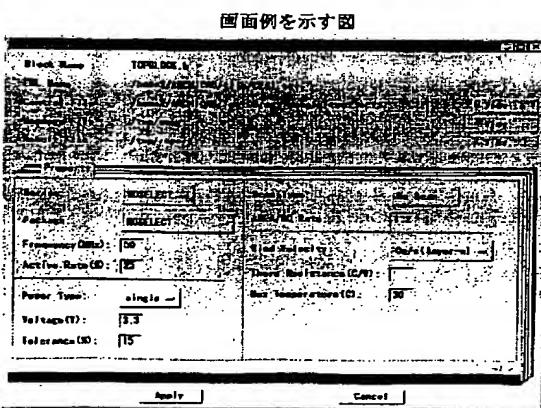


【図26】

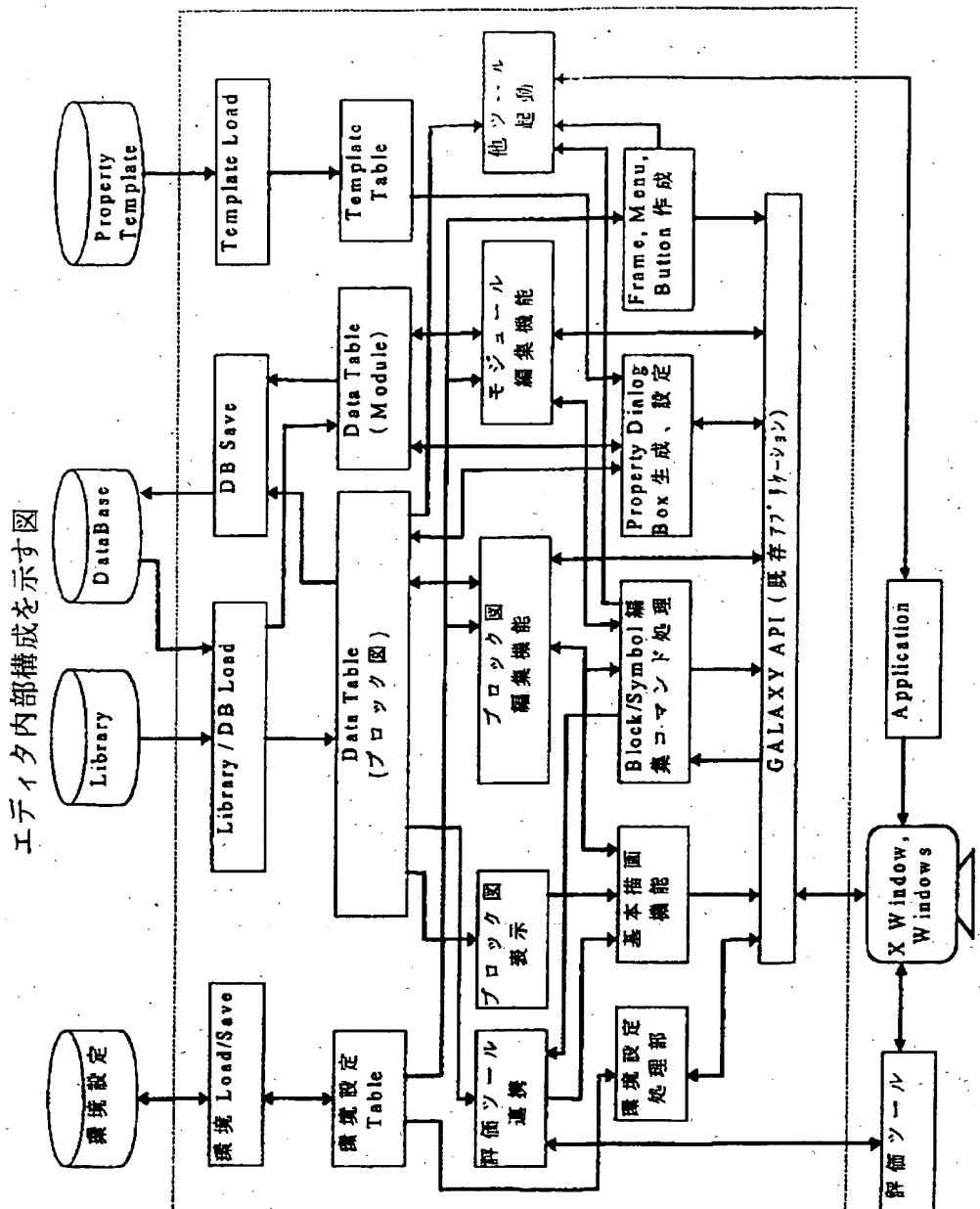
外部テンプレート定義例を示す図

```
# Property Template File (PTF) Define Sample
SYSTEM:
TECHNOLOGY : INPUT S30 MENU(SDICT) : #テクノロジ
#
SYMBOL:
INSTANCE : INPUT S128 DEF(MAUTO) , DISPLAY AL LU 0 10 GREEN : #回転
PARTS : INPUT S20 , DISPLAY AL LU 0 40 YELLOW : #表示用 CPU 等
END:
#
SIGNAL:
TYPE : INPUT S30 MENU(MDICT) , DISPLAY SW CP 0 10 YELLOW :
INTERFACE : FILE DIR(SIFPATH) TYPE(I) :
END:
#
MODULE:
SIZE_BC : INPUT F12 MIN(1) :
COST : INPUT D10 MIN(1) :
END:
#
INTERFACE:
WIRE_NO : INPUT D4 BIN(1) UPDOWN :
RIGHT_SPEED_WIRE : INPUT D4 MAX(WIRE_NO) UPDOWN :
END:
#-- END OF FILE
```

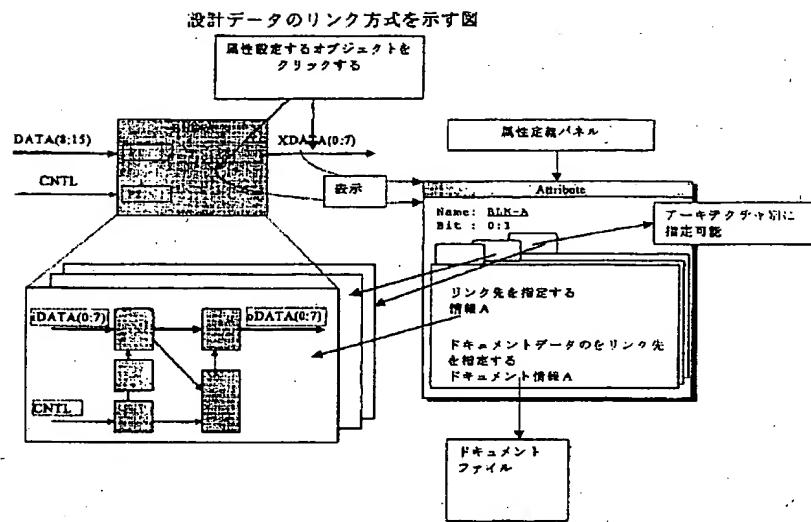
【図27】



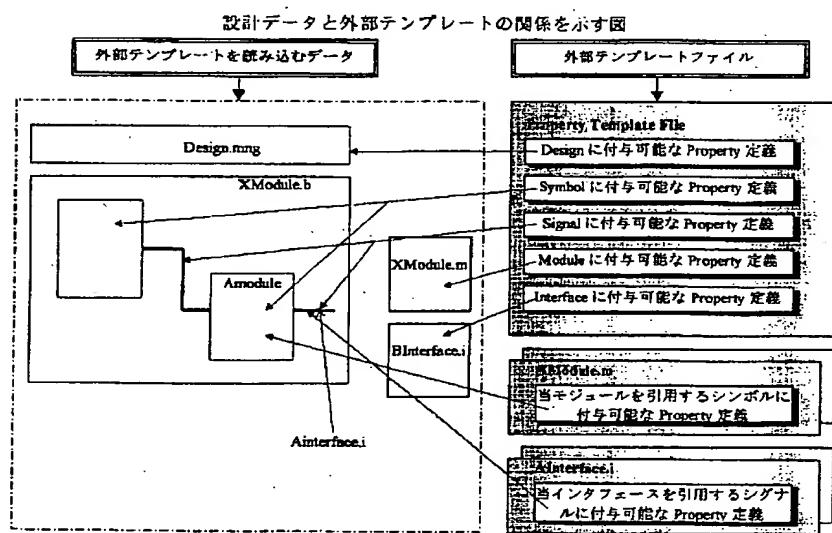
【図21】



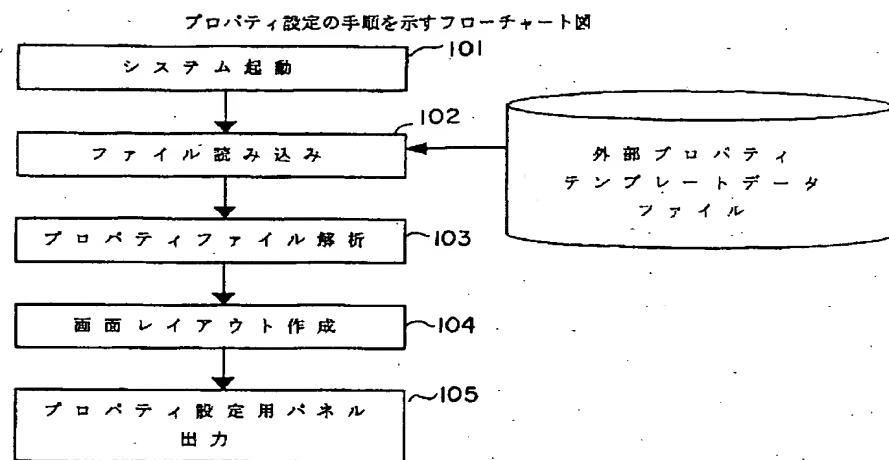
【図23】



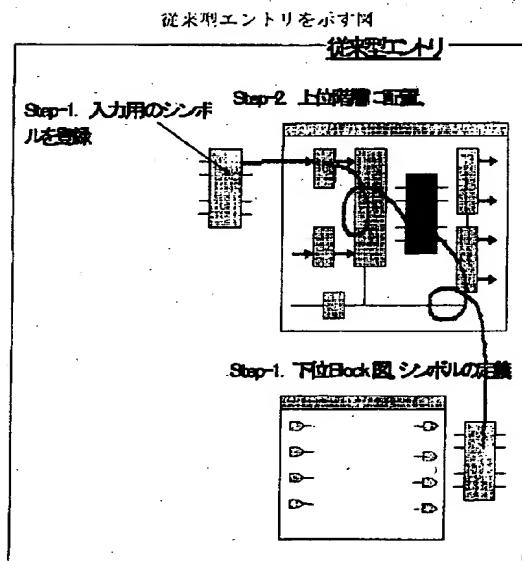
【図24】



【図25】

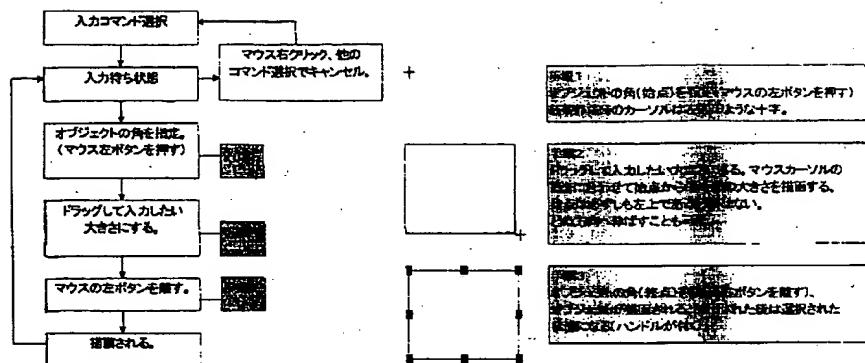


【図28】



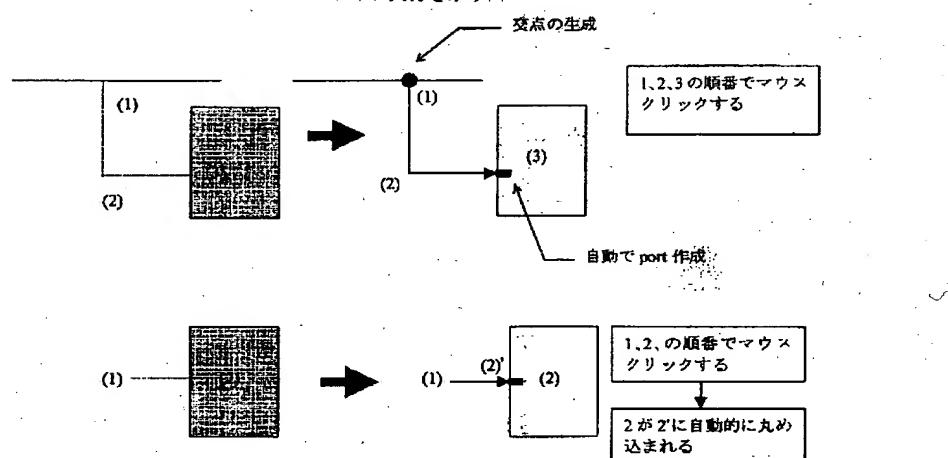
[图30]

シンボル入力フローを示す図



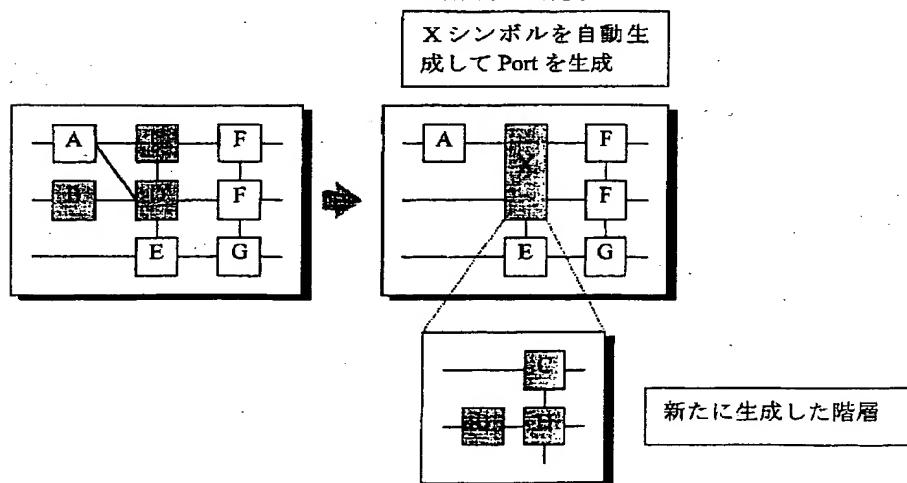
[図31]

ライン入力を示す図

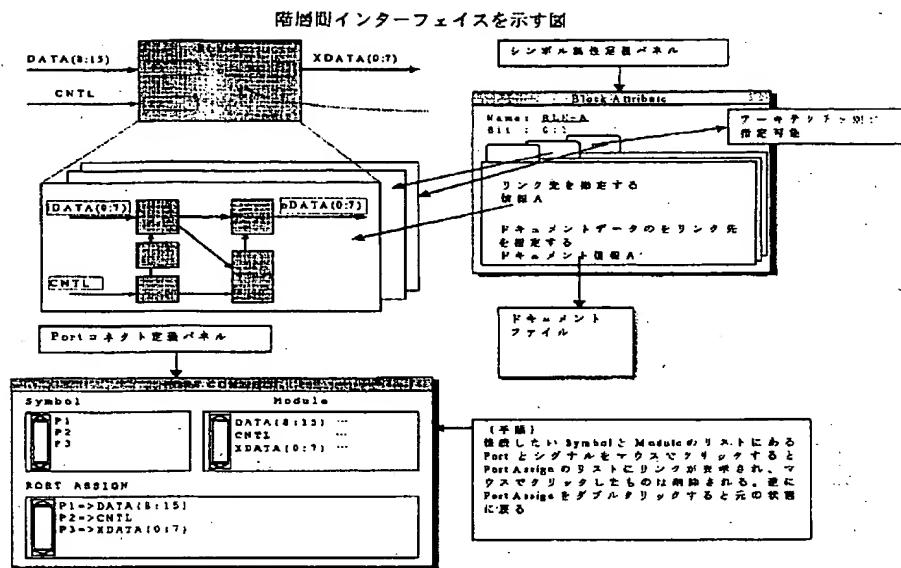


【图3-4】

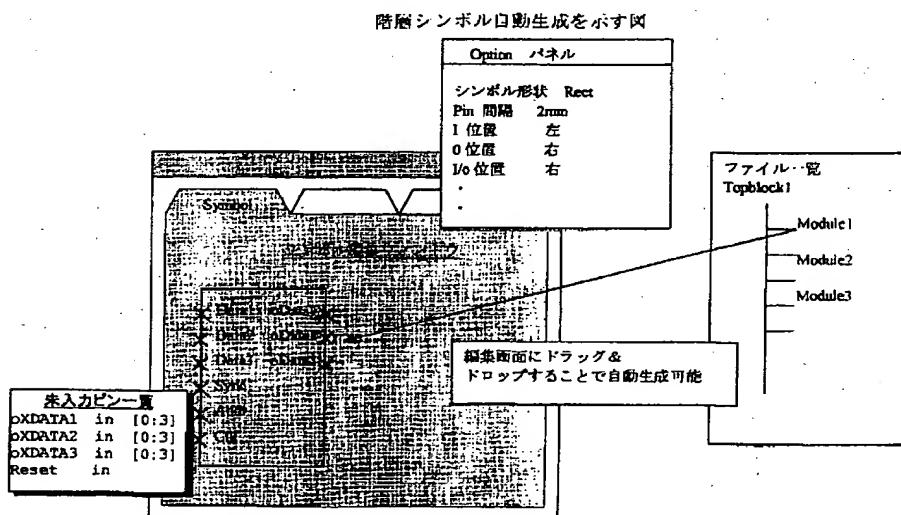
階層展開・合成を示す図



【図32】

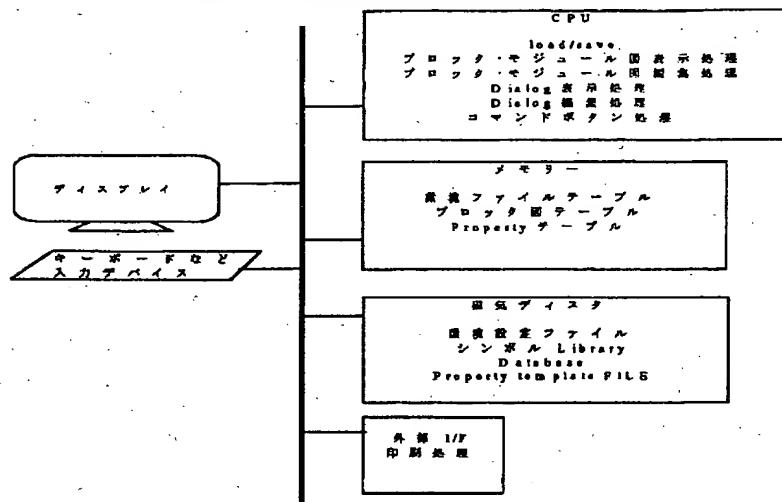


【図33】



【図35】

ソフトウェアで実施した場合の装置構成図を示す図



フロントページの続き

(72)発明者 大塚 正人

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72)発明者 斎藤 孝

栃木県小山市城東3丁目28番1号 富士通
キャドテック株式会社内

(72)発明者 小林 志康

栃木県小山市城東3丁目28番1号 富士通
キャドテック株式会社内

(72)発明者 久保田 健一

栃木県小山市城東3丁目28番1号 富士通
キャドテック株式会社内